HP 3000 Computer Systems

MPE V TABLES MANUAL
MPE V/E (VUF G.08.00)
MPE V Release 23

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.

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(G.03.00) . . . . .June 1987
Second Edition(G.03.05) . . . . November 1988
Third Edition(G.03.05) . . . . April 1989
Fourth Edition(G.03.07) . . . . August 1989
Fifth Edition(G.23.00) . . . . October 1991

Effective Pages   **Date**

ALL . . . . .June 1987
ALL . . . . November 1988
ALL . . . . April 1989
ALL . . . . August 1989
ALL . . . . October 1991

PRINTING HISTORY

New editions are complete revisions of the manual. Update packages, are NOT issued for the MPE V TABLES MANUAL The date on the title page and back cover of the manual changes when a new edition is published.

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.

CONTENTS

CONTENTS (Continued)

CONTENTS (Continued)

CONTENTS (Continued)

CONTENTS (Continued)

CHAPTER 10 PREPARED OBJECT CODE

CONTENTS (Continued)

CONTENTS (Continued)

CONTENTS (Continued)

CHAPTER 14 SPOOLING

CONTENTS (Continued)

CONTENTS (Continued)

CONTENTS (Continued)

CONTENTS (Continued)

CONTENTS (Continued)

CONTENTS (Continued)

CONTENTS (Continued)

CHAPTER 25 ATP/ATP37/ADCC

This edition of the MPE Release 23 Tables Manual describes the internal table organization of the MPE V operating system. It is intended for the technically sophisticated user with Privilege Mode capability. We strongly discourage modifying the contents of the MPE tables 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. Hewlett-Packard reserves the right to change the structure and the content of any system tables in future releases of MPE.

The major highlights of this edition include:

● Corrections/Additions were made in bringing the information up to Release 23.

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 3x/4x/5x/6x/70,Micros)

OCTAL		DEC
0	CSTB (BASE OF CST TABLE)**	0
1	XCSTB (POINTER TO CURRENT EXECUTING PROGRAM BLOCK)	1
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

OCTAL	0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15	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	MEMORY ADDRESS OF FIRST LINKED	
21	MEMORY REGION	
22	TIME OF LAST CYCLE	
23		
24	RESERVED	
25	BREAK POINT FLAG LX SY	BPTF

System Global Area (Cont.)

26	VDSMTAB BASE	VDSMTAB
27	STATIC FENCE (# CONFIGURED MEMORY BANKS)	
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	ABSOLUTE ADDRESS (SYSDIT(8))	DIT8
36	RESERVED	SBANK
37	ABSOLUTE ADDRESS 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	NUMBER OF 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	DISPTOAWAKEMSG
51	CSTBLK TABLE BASE ADDRESS	CSTXBLCKPOINTER
52	PRIORITY OF PROCESS TO BE SERVICED NEXT	AWAKETOSCHEDMSG
53	WAIT --> DISP COMMUNICATION WD	WAITTODISPMSG
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 (0:1) - NO SYSBUF AVAIL FOR I/O ERROR LOGGING (1:1) - NO SYSBUF FOR IOMESSAGE (GENMSG)	IOLOGOX
64	# OF TERMINALS READING	RDCOUNT
65	# OF TERMINALS WRITING	WRTCOUNT
66	DSET B	CRIO
67	LAST TIMER	CRIO
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	
100	MAX # OF SPOOL SECTORS	MAXSSECT
101		

RESERVED FOR I/O SYSTEM (rows 64–77)

RESERVED FOR FILE SYSTEM (rows 100–101)

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 | MODA 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
136	CS FIX LEVEL
137	CS VERSION
\140	CCLOSE PLABEL
141	LOGICAL PROCESS TABLE (PROGEN)
142	LOGICAL PROCESS TABLE (MESSENGER)
143	LOGICAL PROCESS TABLE (UCOP)
144	LOGICAL PROCESS TABLE (PFAIL)
145	LOGICAL PROCESS TABLE (DEVREC)
146	LOGICAL PROCESS TABLE (NMMON)
147	RESERVED
150	LOGICAL PROCESS TABLE (LOG)
151	LOGICAL PROCESS TABLE (LOAD)
152	LOGICAL PROCESS TABLE (IOMESSPROC)
153	LOGICAL PROCESS TABLE (SYSIOPROC)
154	LOGICAL PROCESS TABLE (MEMLOGP)
155	EXTERNAL PLABEL OF "TERMINATE"
156	INTERNAL PLABEL OF "TERMINATE"

RESERVED FOR CS (rows 135–140)

System Global Area (Cont.)

157	EXTERNAL PLABEL OF "COMMANDINTERP"	
160	INTERNAL PLABEL OF "COMMANDINTERP"	
161	EXTERNAL PLABEL OF "SPOOLIN"	
162	INTERNAL PLABEL OF "TRACEO"	
163	EXTERNAL PLABEL OF "TRACEO"	
164	INTERNAL PLABEL OF "SPOOLIN"	
165	EXTERNAL PLABEL OF "SPOOLOUT"	
166	INTERNAL PLABEL OF "SPOOLOUT"	
/167		
170	3 WORD LOGGING MASK	
171		
172	STATE | DST# - BUFFER 0	STATE:
173	STATE | DST# - BUFFER 1	0 EMPTY 1 CUR
174	BUFFER LENGTH (SECTORS)	2 FULL
175	FREE AREA POINTER	
176	FLAGX	
177	# RECORDS WRITTEN IN BUFFER 0	
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	# OF LOGGING BLOCKS WRITTEN (1ST HALF)	
\207	# OF LOGGING BLOCKS WRITTEN (2ND HALF)	

RESERVED FOR LOGGING (rows 177–207)

System Global Area (Cont.)

Addr	Description	Label
/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	
215 216 217	RESERVED FOR KERNEL USE	
220	MAPPING FIRMWARE FLAG BIT 15 V/EXPANSION BIT 14 PMBC BIT 13 S/70	
221 222	BANK / BASE ADDRESS OF MAPPING DST (INITIALIZED BY DISPATCHER DURING LAUNCHING A PROCESS)	
223	NUMBER OF CODE SEGMENTS OF CURRENT PROCESS	
224	TOTAL FREE PHYSICAL CST ENTRIES	
225	HEAD OF FREE PHYSICAL CST LINK	
226	XLST DST NUMBER	
227 247	RESERVED	
250 251	BANK / BASE ADDRESS OF AVAILABLE REGION LIST HEAD	HLHEAD
252 253	BANK / BASE ADDRESS OF AVAILABLE REGION LIST TAIL	HLTAIL

G.23.00
1- 9

System Global Area (Cont.)

Addr	Description	Label
SEGMENT TRACE /254	CURRENT WORD COUNT	XDSCOUNT
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	PERFORMANCE FEATURE SET	
266	MEMORY SCAN POINTER	
267	MEASFLAGS	**
270	HEWLETT-PACKARD DATA BASE (HPDB)	
KERNEL /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)	
275	ADDRESS OF CDT TABLE (DISC CACHING)	
276	HELP LOGICAL DEVICE NUMBER	
\277	CURRENT LOGON DST	DSTLOGON
/300 301	(STOP) (BITS) (see p. 2-15)	
302	# PROCESS ENTRIES	
303		

G.23.00
1- 10

System Global Area (Cont.)

Addr	Description	Value	Label
304	DEVREC PIN	2	
305		X20	
306	UCOP PIN	0	
307		X20	
PROCESS STOP TABLE 310	LOG PIN	1	
311		X20	
312	IOMESS PIN	3	
313		X20	
314	MEMLOG PIN	4	
315		X20	
316	NMMON PIN		
\317		4	
/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	CISC	SIO ERROR
332	RQNESS		
333	MAXQUEUE		JOBPRI
334	DEFAULTQUEUE		

G.23.00
1- 11

System Global Area (Cont.)

Addr	Description	Value	Label
335	DSCHECK PLABEL		
336	DSOPEN PLABEL		
337	DSCLOSE PLABEL		
340	MANAGEWRITE CONV. PLABEL		
341	CONSDSLINE' PLABEL		
342	CXREMOTE PLABEL		
343	CKDSLINE 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	MEM PRESSURE SOFTDEATH FLAG
351 352	LAST CYCLE DURATION		
353 354	CYCLE THRESHOLD		
355	BUG CATCH ENABLE CELL		
356	MONITOR BUFFER	TIMESTAMP	MONBUFT0
357	MONITOR BUFFER	TIMESTAMP	MONBUFT1
360	DSBREAK PLABEL		
361 362	BANK / BASE ADDRESS OF LAST MEMORY WORD		LAST MEMORY ADDRESS

G.23.00
1- 12

System Global Area (Cont.)

```
                 |----------------------------------------|
            /363|           PVPROC PIN                     |
            |    |----------------------------------------|
            |364 |        PV RECOGNITION COUNT             |
PRIVATE  <  |    |---------------------------|--|--|--|--| |
VOLUMES  |365|          VMOUNT FLAGS         |MT|AU|AL|ON| |
            |    |---------------------------|--|--|--|--| |
            |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 SOFTDUMP )       |
                 |----------------------------------------|
           /401 |       MICROCODE MEMORY LOCATIONS        |
            |    |                                        |
            |    | *NOTE THAT THE CONTENTS DEPEND ON THE TYPE|
            |    | OF CPU THAT MPE IS RUNNING AND WHETHER A  |
            |    | DUMP, POWERFAIL, OR CNTL B/HALT HAS OCCURRED|
            \402 |----------------------------------------|
```

** MEASFLAGS (12:1) = 1 = Tape Error   VMOUNT Flags: MT - mount
               (13:1) = 1 = EOT on Monitor Tape   AU - auto
               (14:1) = 1 = Buffer Flip/Flop   AL - all
               (15:1) = 1 = Monitor Enabled   ON - on

The following locations refer to all systems:

%1401 = DUMPDEVDRT	%1406 = Q	%1413 = PB - BANK
1402 = X	1407 = S	1414 = PB
1403 = DL	1410 = S - BANK	1415 = P
1404 = DB - BANK	1411 = Z	1416 = PL
1405 = DB	1412 = STATUS	1417 = CIR
		1420 = High Bank

The following locations refer exclusively to the Series 37:

%1421 = Microcode Version Number
        Bit (0:2) 00 = Master Released
                   10 = Pending Release
                   11 = Experimental
        Bit (2:6) Base Level (1-64)
        Bit (8:8) Patch Level (1-99)
%1422 = Flags/Misc
        BIT (0:1) 1 If On ICS
        Bit (1:1) 1 If In Dispatcher
        Bit (2:1) Logical/Physical
                   1 If Logical
        Bit (3:1) 1 If Channel Program Is Running
        Bit(4:1)   Split Bank Flag
                   1 If Split
        Bit(5:3)   Unused
        Bit(8:8)   Last Stop Code
%1423/7377 = Channel Program Area For Booting Software
             (Used Only During Boot).

The following are assignments after software has been loaded and launched:

%1540/1617 = ROM Input Buffer For Terminal I/O
 1620/1677 = ROM Output Buffer For Terminal I/O
 1700/1710 = ROM Control Buffer For Terminal I/O
 1711/1737 = ROM Control B Interface Buffers

The following assignments refer to the Series 3x/4x/5x/6x/70,Micros

30/33/39/4x/5x	6x/70
%1421 = System Halt #	%1421 = CPX1 Register
1422 = ISR (Interrupt Register)	1422 = CPX2 Register
	1514 = System Halt Flag
%1515 = System Interrupt Mask	%1515 = NIR Register
1516 = DRT 0	
1517 = DRT 1	37/6x/70,Micros
1520 = DRT 2	
1521 = DRT 3	1516 = DRT 0
	1517 = DRT 1
	1520 = DRT 2
	1521 = DRT 3
	1522 = DRT Bank
	1523 = DRT Address Offset
	(6x/70 only) 1524 = Interrupt Mask For IMB0
	(6x/70 only) 1525 = Interrupt Mask For IMB1
	(6x/70 only) 1526 = Interrupt Mask For IMB2
	(6x/70 only) 1527 = Interrupt Mask For IMB3

All Systems: 1740 = Start Of SysGlob Extension

System Halt Flag: 0 - Unexpected (unknown) Interrupt
                   1 - STT Violation in Segment #1
                   2 - Absent Segment while executing on the ICS
                   3 - Absent or Trace on Segment #1
                   4 - Stack Overflow on the ICS
                   5 - CST Table length (word 0) found to be 0
                   6 - Bootstrap Channel Program Timeout
                   7 - Bootstrap Checksum Error
                   8 - Bootstrap Channel Program Abort
                   9 - PSEB Instruction while QI-18 = 0
                  10 - Module Send Message Timeout
                  11 - Incorrect Module Responding
                  12 - Channel not System Controller
                  13 - Code Segment Violation while in Segment #1
                  14 - Non-responding Channel
                  15 - Channel 0 Responding (to IPOLL)
                  16 - No CSRQ or IRQ on Message Interrupt
                  17 - Channel cannot be made Controller-in-charge
                  18 - Module Receive Message Timeout
                  19 - I/O Error, Parity or Timeout
                  20 - UCS Checksum Error
                  21 - LUT Checksum Error
                  22 - Bad CPU Command Code

SysGlob Extension

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

```
          |----------------------------------------|
     % 0| |     SWAP QUEUE DELAY (*100MS)          |SWAPQDELAY
          |----------------------------------------|
       1| |     BANK / BASE ADDRESS                |FIRST
          |       OF FIRST MEMORY                   |MEMORY
       2| |     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| |                                        |HOTIMELAST-
          |   TIME OF LAST MAKEROOM CALL           |MAKEROOM
          |                                        |LOTIMELAST-
      10| |                                        |MAKEROOM
          |----------------------------------------|
      11| |   MEMORY PRESSURE DURATION THRESHOLD   |
          |----------------------------------------|
      12| |   NATIVE LANGUAGE TABLE (NLT) DST #    |
          |----------------------------------------|
      13| |   RESERVED FOR NATIVE LANGUAGE SUPPORT |
          |----------------------------------------|
      14| |   BAUD RATE OF THE SYSTEM CONSOLE      |
          |----------------------------------------|
      15| |   INITIAL LDEV OF SYSTEM CONSOLE       |
          |----------------------------------------|
      16| |   PLABEL FOR REMOTE'MPE               |
          |----------------------------------------|
      17| |   PLABEL FOR GETDS'NODENAME           |
          |----------------------------------------|
      20| |   UCS VERSION                          |
          |----------------------------------------|
      21| |   WORD 0 MICROCODED PROCEDURES,        |
          |       PERFORMANCE FEATURE SET BIT MAP  |
          |----------------------------------------|
      22| |   WORD 1 MICROCODED PROCEDURES,        |
          |       PERFORMANCE FEATURE SET BIT MAP  |
          |----------------------------------------|
      23| |   WORD 2 MICROCODED PROCEDURES,        |
          |       PERFORMANCE FEATURE SET BIT MAP  |
          |----------------------------------------|
      24| |   WORD 3 MICROCODED PROCEDURES,        |
          |       PERFORMANCE FEATURE SET BIT MAP  |
          |----------------------------------------|
```

SysGlob Extension (Cont.)

For Word 3 (%24) the following applies:

Bit 15 is set if ERRORON'70 is present
Bit 14 is set if ERROREXIT'70 is present
Bit 13 is set if EXCHANGEDB'70 is present
Bit 12 is set if TIMER'70 is present
Bit 11 is set if TIMEREQ'70 is present
Bit 10 is set if MMSTAT'70 is present

```
      |-------------------------------------|
  30| |  SECURITY TABLE                     |
      |                                     |
      |                                     |
  56| |-------------------------------------|
  57| |                                     |
      |-------------------------------------|
  60| |  PLABEL USERLOG (EXTERNAL)          |
      |-------------------------------------|
  61| |  PLABEL USERLOG (INTERNAL)          |
      |-------------------------------------|
  62| |  PLABEL RECLOG  (EXTERNAL)          |
      |-------------------------------------|
  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        | MIOCNT |*
      |-------------------------------------|
  72| |  LOADER CACHE SEGMENT NUMBER        |
      |-------------------------------------|
  73| |  PLABEL 3270    (EXTERNAL)          |
      |-------------------------------------|
  74| |  VERSION                            |
      |-------------------------------------|
  75| |  UPDATE                             |
      |-------------------------------------|
  76| |  FIX                                |
      |-------------------------------------|
```

G.23.00
1- 17

SysGlob Extension (Cont.)

```
      |-------------------------------------|
  77| |  COUNT OF TAPE CONTROLLERS USING MEASIO |
      |-------------------------------------|
 100| |  PORT DATA SEGMENT NUMBER           |
      |-------------------------------------|
 101| |  RESERVED FOR SECOND PORT DATA SEGMENT |
      |---                                  |
   v | SYSTEM FPMAP OPTION FLAG             | SYSFPMAP
      |-------------------------------------|
 103| |                                     |
 104| |                                     |
 105| |                                     |
 106| |  GLOBAL ALLOW MASK                  |
 107| |                                     |
 110| |                                     |
      |-------------------------------------|
 111| |  MMSTAT ENABLE WORD                 | ***
      |-------------------------------------|
 112| |  RESERVED                           |
      |                                     |
 117| |                                     |
      |-------------------------------------|
 120| |  SYS PORT PROCESS PCB RELATIVE INDEX |
      |-------------------------------------|
 121| |  GLOBAL AFT DST NUMBER              |
      |-------------------------------------|
 122| |  INITIAL/PROGEN COMM. DSEG NUMBER (CH. 16) |
      |-------------------------------------|
 123| |  INITIAL SYSTEM STARTUP OPTION      |
      |-------------------------------------|
 124| |  PORT'MAX'SER'COUNTER               |
      |-------------------------------------|
 125| |                                     |
 126| |  CURRENTLY UNASSIGNED               |
 127| |                                     |
      |-------------------------------------|   SYSGLOB EXT.
 130| |  Address Allocation DST             |   %130 - %144
      |-------------------------------------|   HAVE BEEN
 131| |  IPC Plabel                         |   ALLOCATED TO
      |-------------------------------------|   NS TRANSPORT
 132| |  Multicast DST                      |
      |-------------------------------------|
 133| |  PD DST                             |
      |-------------------------------------|
 134| |  IP Update DST                      |
      |-------------------------------------|
 135| |  Node Name DST                      |
      |-------------------------------------|
 136| |  Address Reference DST              |
      |-------------------------------------|
 137| |  IP Identification                  |
      |-------------------------------------|
```

G.23.00
1- 18

SysGlob Extension (Cont.)

```
      |-------------------------------------|
 140| |  SM4 DST                            |
      |-------------------------------------|
 141| |  Flags word                         |
      |-------------------------------------|
 142| |  Trigger DST                        |
      |-------------------------------------|
 143| |  Util. DST                          |
      |-------------------------------------|
 144| |  FastPath SM4                       |
      |-------------------------------------|
 145| |  RESERVED FOR SPL                   |
      |-------------------------------------|
 146| |  PATH FLOW                          |
      |-------------------------------------|
 147| |  ANALYZER                           |
      |                                     |
 150| |                                     |
      |-------------------------------------|
 151| |  PC SUBSYSTEM INFO                  |
      |-------------------------------------|
 152| |  KODIAK M-CODE PERFORMANCE          |
      |-------------------------------------|
 153| |  MAESTRO'WORD                       |
      |-------------------------------------|
 154| |  MAESTRO DST-NUMBER                 |
      |-------------------------------------|
 155| |  << SYSGLOB %155 IS UNUSED >>       |
      |-------------------------------------|
 156| |  PSEUDO TERMINAL LINKAGE            |
      |-------------------------------------|
 157| |  VIRTUAL TERMINAL LINKAGE           |
      |-------------------------------------|
 160| |  DST # FOR NS/3000, X.25 PRODUCT STARTED |
      |-------------------------------------|
 161| |  PAD (Package Assembly/Disassembly) |
      |-------------------------------------|
 162| |  MPE PRODUCT VERSION LEVEL (VD4 and later) |
      |-------------------------------------|
 163| |  MPE PRODUCT UPDATE LEVEL (VD4 and later) |
      |-------------------------------------|
 164| |  MPE PRODUCT FIX LEVEL (VD4 and later) |
      |-------------------------------------|
 165| |  NS/X.25 DST # (VD6 and later)      |
      |-------------------------------------|
 166| |  PLABEL OF TAPE MGMT CATALOG PROCEDURE |
      |-------------------------------------|
 167| |  RESERVED FOR KNOWN VENDOR TABLE    |
      |                                     |
 200| |                                     |
      |-------------------------------------|
```

G.23.00
1- 19

```
      * MIOCNT = MEASIOCOUNT  (3 BITS)

    *** BIT 0 = Enable HARDRES MMSTAT call

        MAESTRO'WORD DEFINITION
                    (0:1) = 1 = START'UP'BIT
                    (1:9)   -   (Not Used)
                   (10:1) = 1 = Pending Spooler Request
                   (11:1) = 1 = Queue Entry in XDS

        MAESTRO DST-NUMBER - Contains the DST index for the XDS
                             shared between Filter and GENMSG.

        NS/X.25 DST#(word %165) - DST # for the MPE V/E NS/X.25
                                  TRANSPORT link for INP's.
        STARTUP OPTION(word %123) - Contains the last LOAD/START
                                    option as follows:

                             0 = WARMSTART
                             1 = COOLSTART
                             2 = COLDLOAD(or COLDSTART)
                             3 = UPDATE
                             4 = RELOAD (ACCTS/NULL/SPREAD)
                             5 = RELOAD (COMPACT)
                             6 = RELOAD (RESTORE)
```

G.23.00
1- 20

SYSDB Words

Systen 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	CONSLDEV	- 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

Allow Mask (Cont.)

	Word	Bit	#
UPPER LIMIT->DEVICE COMMANDS			
ABORTJOB	0	13	13
ALLOW	0	14	14
ALTSPOOLFILE	0	15	15
ALTJOB	1	0	16
BREAKJOB	1	1	17
DELETESPOOLFILE	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
DOWNLORD	2	7	39
MIOENABLE	2	8	40
MIODISABLE	2	9	41
LOG	2	10	42
FOREIGN	2	11	43
IMFCONTROL	2	12	44
SHOWCOM	2	13	45
OPENQ	2	14	46
SHUTQ	2	15	47
DISCRPS	3	0	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 DRTs 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 DRTs 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 6x/70 if
|------------------------------|   Privilege Mode Bounds
| SYSTEM GLOBAL AREA           |   Checking is enabled.)
|------------------------------|
| FIRMWARE AREA                |
|------------------------------|
| SYSGLOB EXTENSION            |
|------------------------------|
| DST/CST/CSTX                 |
|------------------------------|
| ICS                          |
|------------------------------|
| PMBC                         |   (Only for 6x/70 if
|------------------------------|   Privilege 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                         |
|------------------------------|
```

Bank 0 (Cont.)

```
|------------------------------|
| >MON TABLE                   |
|------------------------------|
| CORE RESIDENT CST'S          |
|    IN 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                 |
|-----------------------------------|
```
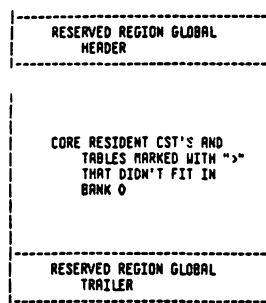
CHAPTER 2 MEMORY MANAGEMENT TABLES

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 below. 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.

Overall ST Structure

```
(X2),(X1002)+SYSBASE --------> |-------|
                               |       |
                               |  DST  |
(X0),(X1001)+SYSBASE --------> |-------|
                               |       |
                               |  CST  |
                               |       |
                    CSTBLK     |-------|  |-------> |-------|
                    |-------|  |       |            | CSTX  |
(X1051) -->|        |       |  |       |            | FIRST |
           |        |-------|  |       |            | LOADED|
           |        |       |  |       |            |PROGRAM|
           |        |-------|  |----|  |-------> |-------|<-- (X3), CURRENT PROGRAM POINTER
           |        |       |       |            | NEXT  |
           |        |       |       |            | LOADED|
           |        |       |       |            |PROGRAM|
           |        |       |       |            |-------|
           |        |-------|       |            |       |
           |        |       |       |            |       |
           |        |-------|  |-------|          |       |
           |        |       |                     |-------|
           |        |       |
           |        |-------|
```
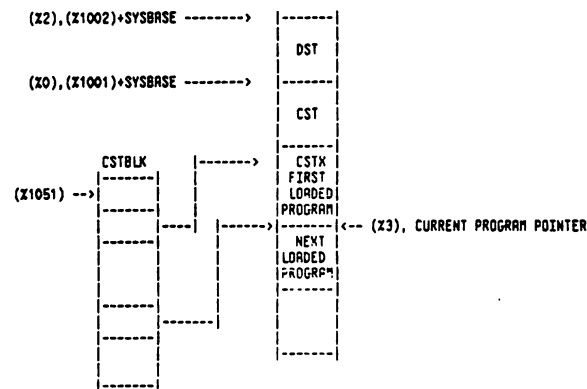
Pointers and DST #'s of Segment Table Components

    i. DST

        X 2 absolute address of entry 0 of the DST.
        X1002 sysbase relative index of entry 0 of DST.
        DST number 2 is the DST Table DST #.

    ii. CST

        X 0 absolute address of entry 0 of System SL.
        X1001 sysbase relative index of entry 0 of System SL.
        X1032 displacement from DST base of entry 0 of System SL (i.e.,
        @CST(last) - @DST(0) = DFS).
        DST number 1 is the CST Table DST #.

    iii. CSTX

        X 1 absolute address of entry 0 of current program.
        X1033 displacement from DST base to first CSTX entry SL.
        DST number 4 is the CSTX Table DST #.

    iv. CSTBLK

        X1051 sysbase relative index of CST Block Table.
        DST number 35 (X43) is CSTBLK's DST #.

Standard Object Identifier Format

```
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
|   TYPE   |     CST BLOCK - RELATIVE INDEX      |
|-----------------------------------------------|
|              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) = CST BLOCK Table index to list of CST segments
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 |N |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 |N |T |O |W |Y |O |D |    VMALLOC     |FLAGS
      |V |C |I |K |D |I |S |R |   |                |
      |  |  |  |  |  |P |  |E |   |                |
          |--------------------------------------------|
WORD 2|    LDEV #              |     MODA         |MODA
          |--------------------------------------------|
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 #              |     MODA         |MODA
          |--------------------------------------------|
WORD 3|    LODA                                   |LODA
          |--------------------------------------------|
```

Case (iii) DST/CST Free Entry

```
          |--------------------------------------------|
          |    Z100000                                 |
          |--------------------------------------------|
          |    TABLE RELATIVE OFFSET TO NEXT FREE ENTRY|
          |--------------------------------------------|
          |                                            |
          |--------------------------------------------|
          |                                            |
          |--------------------------------------------|
```

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

CST 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 change in size or location is requested
    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

```
            0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
            |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
CSTBLK 0|    NUMBER OF ENTRIES IN CSTBLK TABLE    |
            |----------------------------------------|
        1|    ENTRY NUMBER 1                       |
            |----------------------------------------|
        2|    ENTRY NUMBER 2                       |
            |----------------------------------------|
        3|    ENTRY NUMBER 3                       |
            |----------------------------------------|
         |      :            :            :         |
         |      :            :            :         |
         |      :            :            :         |
            |----------------------------------------|
        m|    ENTRY NUMBER m                       |
            |----------------------------------------|
```

The table entries are initialized to -1 to denote an unassigned entry. When an entry is assigned, its contents is replaced with a DST relative address which points to the header entry of the code segment list (see the CST EXTENSION table format for more information).

CST EXTENSION 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. The CSTXMAP is a contiguous block of entries inside the CST EXTENSION. It contains a header entry describing the block of entries and a group of CST entries describing each of the program code segments. The start of the CSTXMAP is pointed to by an entry (CSTXEIX) from the CST BLOCK (CSTBLK) table.

Entry Format - CSTXMAP

```
              0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
              |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
HEADER  0|    # OF CODE SEGMENT ENTRIES OF THIS BLOCK |
              |-------------------------------------------|
        1|    Z125252                                 |
              |-------------------------------------------|
        2|    NUMBER OF USERS SHARING THIS BLOCK       |
              |-------------------------------------------|
        3|    0                                       |
              |--|--|--|--|-------------------------------|
CODE SEG 1+0|A |P |R |T |    SIZE/4 (WORDS)            |
              |-------------------------------------------|
         |D |R |I |S |M |F |S |C |W |                 |
        1|C |O |N |T |O |W |Y |O |D |      VMALLOC     |
         |V |C |I |K |D |I |S |R |   |                 |
         |  |  |  |  |  |P |  |E |   |                 |
              |-------------------------------------------|
        2|    BANK/BASE ADDRESS OF                     |
         |    CODE SEGMENT IN MEMORY                   |
         |              OR                             |
        3|    LDEV# MODA/LODA ADDRESS                  |
         |    OF CODE SEGMENT ON DISC                  |
              |--|--|--|--|-------------------------------|
CODE SEG 2+0|A |P |R |T |    SIZE/4 (WORDS)            |
              |-------------------------------------------|
         |D |R |I |S |M |F |S |C |W |                 |
        1|C |O |N |T |O |W |Y |O |D |      VMALLOC     |
         |V |C |I |K |D |I |S |R |   |                 |
         |  |  |  |  |  |P |  |E |   |                 |
              |-------------------------------------------|
        2|    BANK/BASE ADDRESS OF                     |
         |    CODE SEGMENT IN MEMORY                   |
         |              OR                             |
        3|    LDEV# MODA/LODA ADDRESS                  |
         |    OF CODE SEGMENT ON DISC                  |
              |-------------------------------------------|
```

Entry Format - CSTXMAP (Cont.)

```
         |--------------------------------------------|
         |  :              :                          |
         |  :              :                          |
         |  :              :                          |
         |  :              :                          |
         |--|-|--|-|----------------------------------|
CODE SEG n+0|A |P |R |T |      SIZE/4 (WORDS)          |
         |--------------------------------------------|
         | |D |I |I |S |M |F |S |C |W |               |
        1|C |O |M |T |O |W |Y |O |D |   VMALLOC       |
         | |V |C |I |K |D |I |S |R |   |               |
         | | |I | | |P | |E | |               |
         |--------------------------------------------|
        2|     BANK/BASE ADDRESS OF                   |
         |         CODE SEGMENT IN MEMORY             |
         |                  OR                        |
        3|     LDEV# HODA/LODA ADDRESS                |
         |         OF CODE SEGMENT ON DISC            |
         |--------------------------------------------|
```

The value of CSTXEIX is established when a CST extension block is allocated.
This index into the array CSTBLK 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

* Can be moved out of BANK 0 if necessary.

Fixed DST Entry Assignments (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	

* Can be moved out of BANK 0 if necessary.

Fixed DST Entry Assignments (Cont.)

OCTAL		DECIMAL	TABLE NAME
52	ILT	42	ILT
53	*SIR TABLE	43	SIR
54	FMRVT	44	FMRVT
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	CILOGOST
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	

* Can be moved out of BANK 0 if necessary.

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 of the process.

    SWAPTAB DST# = 23 (X27)

    X1004 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

OCTAL	0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15	DECIMAL
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
10	TAIL OF IMPEDED QUEUE (PCB RELATIVE)	8
11	# CURRENTLY IMPEDED PROCESSES	9
12	MAX # OF IMPEDED PROCESSES	10
13	CUMULATIVE # OF IMPEDED PROCESSES	11
14	.	12
	.	
	.	
21		17

SWAPTAB Unassigned Entry Format

OCTAL	0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15	DECIMAL
0	X100000	0
1	TABLE RELATIVE INDEX OF NEXT FREE ENTRY	1
2	TABLE RELATIVE INDEX OF PREV FREE ENTRY	2
3	0	3
4	0	4
5	0	5

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

Notes:

    Word 0: In an unused entry only has X100000 if this entry was
             previously for a DST or CST, otherwise it is 0.

    Word 2: The PREVIOUS pointers are not valid. MPE does not maintain
             (or use) them. Only NEXT pointers are valid.

    Words 3-5: Are not zeroed out when a used entry becomes free, but
                will still contain the old data. They are only zero'd
                when the table is first initialized.

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 |    |----------------------|
```
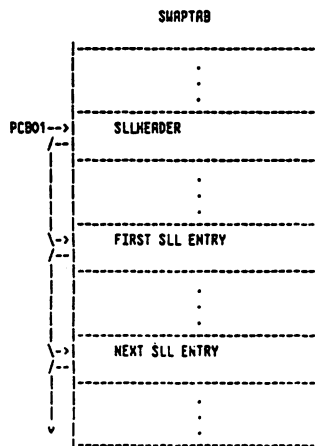
Segment Locality List (SLL) Header Format

```
    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 |                    |
   |R |S |T |R |T |I |      IOCNT         |SCHEDTOIOMSG
 0 |E |M |L |T |T |P |                    |
   |Q |E |O |I |I |  |                    |
   |  |M |C |N |V |  |                    |
   |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
 1 | THREAD TO FIRST SLL ENTRY          |FIRSTINX
 2 |------------------------------------|
 3 | SLL REL ADDRESS OF OBJECT TO BE    |MEMREQINX
   | BROUGHT IN                         |
   |------------------------------------|
 4 | # OF SEGMENT LOCALITY LIST ENTRIES |SEGCOUNT
   | OF PROCESS                         |
   |------------------------------------|
 5 |                                    |
   |------------------------------------|
```

SLL(SLLHEADINX+0)
        .(1:1) SWREQ, Swap Required Flag
        .(2:1) HASMEM, Has Memory Flag
        .(3:1) INTLOC, Initialize locality list 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, Number of READ I/O completions until
                SWAPIN is completed and the process
                is able to be awakened

Segment Locality List (SLL) 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      |NEXTIMPPIN
    |           WAITING PIN               |
     |-------------------------------------|
   1| SLL REL ADDRESS TO THE NEXT ENTRY   |NEXTINX
    |           IN THE LIST               |
     |-------------------------------------|
   2| SLL REL ADDR TO THE PREVIOUS ENTRY  |PREVINX
    |           IN THE LIST               |
     |-------------------------------------|
   3|  OBJECT                             |SLL'OBJDESC
    |-                                   -|
   4|  IDENTIFIER                         |SLL'OBJNUM
     |-------------------------------------|
    |M |S |D |L |B |F |S |T |F |L |D |        |
    |A |T |I |O |L |R |L |O |Z |K |E |        |
   5|P |R |S |C |K |O |L |S |R |R |C | PREFETCH |SLL'FLAGS
    |S |C |C |K |R |Z |I |S |E |E |C |  COUNT  |
    |E |K |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, Decrement # I/O completion before
                       awake flag
    .(11:5) PREFETCHCOUNT, Number of prefetch segment
                       request counter

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

   * See Standard Object Identifier Format for more information.

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.

Z1042 - SRT relative index to entry # 0
Z1043 - 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 FIRST         |
    |        FREE ENTRY                  |
    |-----------------------------------|
  4|  TABLE REL. INDEX OF LAST          |
    |        FREE ENTRY                  |
    |-----------------------------------|
  5|  HIGH WATER MARK                   |
    |-----------------------------------|
  6|  # PRIMARY ENTRIES                 |
    |-----------------------------------|
  7|  HEAD OF IMPEDED QUEUE             |
    |        (PCB REL.)                  |
    |-----------------------------------|
 10|  TAIL OF IMPEDED QUEUE             |
    |        (PCB REL.)                  |
    |-----------------------------------|
 11|  # CURRENTLY IMPEDED PROCESSES     |
    |-----------------------------------|
 12|  # MAXIMUM IMPEDED PROCESSES       |
    |-----------------------------------|
 13|  CUMULATIVE # OF IMPEDED           |
    |        PROCESSES                   |
    |-----------------------------------|
 14|                                    |
    |              .                     |
    |              .                     |
    |              .                     |
 21|                                    |
    |-----------------------------------|
```

SRT Entry 0 Format (Cont.)

The following entry format is for data segment size/location modifications:

```
    |-----------------------------------|
  0|   NEXT ENTRY FOR DATA SEGMENTS     |
    |-----------------------------------|
  1|   OBJECT                           |
    |-                                 -|
  2|   IDENTIFIER                       |
    |-----------------------------------|
  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.

The following is the format for a request to have the data segment moved:

```
   |--------------------------------------|
 0 |        THREAD TO NEXT ENTRY           |
   |--------------------------------------|
 1 |                                      |
   |--------------------------------------|
 2 |                                      |
   |--------------------------------------|
 3 |            NEW SIZE                   |
   |--------------------------------------|
 4 |      STARTING SOURCE ADDRESS         |
   |--------------------------------------|
 5 |          MOVE LENGTH                 |
   |--------------------------------------|
```

Main Memory Region Headers and Trailers

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

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" (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.)

Global Region Trailer

```
         0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
         |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
|RB-X34|            NOT USED                        |
         |---------------------------------------------|
|RB-X33|    TRAILER SUBREGION SIZE (PAGES)           |PTSS
         |---------------------------------------------|
|RB-X32|    TRAILER REGION STATE                     |PTRAS
       | |A |R |A |C |                               |
       | |S |E |V |L |                               |
       | |S |S |  |N |                               |
         |---------------------------------------------|
|RB-X31|    TRAILER REGION SIZE (PAGES)              |PTRS
         |---------------------------------------------|
```

Trailer length = 4

Global Region Header (Available Regions)

```
         0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
         |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
|RB-X30|    REGION ASSIGNMENT STATE                  |RRAS
       |A |R |A |C |S |L |F |I |I |L |            |M |
       |S |E |V |C |C |Z |Z |O |S |            |I |
       |S |S |  |N |  |P |N |F |T |            |P |
       |  |  |  |D |  |  |  |  |Z |T           |  |
       |  |  |  |  |  |  |  |  |N |            |  |
         |---------------------------------------------|
|RB-X27|    REGION SIZE IN PAGES                     |RS
         |---------------------------------------------|
|RB-X26|                                            |
         |---------------------------------------------|
|RB-X25|                                            |
         |---------------------------------------------|
|RB-X24|   PREVIOUS LINK  (ADDRESS OF PL FIELD       |PL
       |-                                           -|
       |        OF PREVIOUS AVAILABLE REGION)        |
         |---------------------------------------------|
|RB-X22|   NEXT LINK  (ADDRESS OF NL FIELD)          |NL
       |-                                           -|
       |        IN NEXT AVAILABLE REGION)            |
         |---------------------------------------------|
|RB-X20|                                            |
         |---------------------------------------------|
```

Header length = 24 (X30)

Subregion Header (Available Regions)

```
         0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
         |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
|RB-X17|   SUBREGION ASSIGNMENT STATE          |I   |SRS
       |C |R |R |R |R |                        |0   |
       |A |E |O |E |E |                        |     |
       |C |F |C |F |F |                        |S   |
       |H |1 |  |2 |3 |                        |T   |
         |---------------------------------------------|
|RB-X16|   SUBREGION SIZE IN PAGES                   |SS
         |-|-------------------------------------------|
|RB-X15|V| SUBREGION DISPLACEMENT IN MAIN MEM. PAGES  |SD
         |-|-------------------------------------------|
|RB-X14|   WRITE REQUEST POINTER                     |WREQP
         |---------------------------------------------|
|RB-X13|   OBJECT                                    |OBJIDENT
       |-                                           -|
       |          IDENTIFIER                         |
         |---------------------------------------------|
|RB-X11|                                            |
         |---------------------------------------------|
|RB-X10|                                            |
         |---------------------------------------------|
| RB-X7|   LDEV          |        HODA               |HODA
         |----------------|----------------------------|
| RB-X6|   LOW ORDER DISC ADDRESS                    |LODA
         |---------------------------------------------|
| RB-X5|                                            |
         |---------------------------------------------|
| RB-X4|                                            |
         |---------------------------------------------|
| RB-X3|                                            |
         |---------------------------------------------|
| RB-X2|                                            |
         |---------------------------------------------|
| RB-X1|                                            |
         |---------------------------------------------|
```

Global Region Header (Reserved Regions)

```
         0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
         |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
|RB-X30|   REGION ASSIGNMENT STATE                   |RRAS
       |A |R |A |C |S |L |F |I |I |L |M |        |B |
       |S |E |V |L |I |O |Z |O |S |A |        |L |
       |S |S |  |Z |Z |C |N |F |T |P |        |M |
       |I |  |  |I |A |E |K |  |Z |T |        |I |
       |G |  |L |N |  |  |  |  |N |  |        |P |
         |---------------------------------------------|
|RB-X27|   REGION SIZE IN PAGES                      |RS
         |---------------------------------------------|
|RB-X26|   ON GOING I/O COUNT                        |IOCNT
         |---------------------------------------------|
|RB-X25|   INITIATION MESSAGE                        |INITMSG
       |T |E |O |Q |I |E |G |M |R |M |        |M |
       |O |X |N |U |N |X |A |I |E |S |        |S |
       |G |T |G |E |P |R |G |L |G |  |        |G |
       |G |D |O |S |O |R |B |A |P |S |        |V |
       |L |I |I |E |R |E |A |B |A |T |        |A |
       |E |S |N |G |Q |G |G |G |R |  |        |L |
       |  |A |G |R |S |U |E |R |E |R |        |I |
       |  |B |D |V |E |  |R |E |T |  |        |D |
       |  |L |  |E |  |  |  |T |  |  |        |  |
         |---------------------------------------------|
|RB-X24|   ORQ REL ENTRY ADDRESS                     |INITINFO
         |---------------------------------------------|
|RB-X23|   COMPLETION MESSAGE                        |COMPMSG
       |T |M |B |S |I |M |                    |M |
       |O |O |L |C |O |S |                    |S |
       |G |V |K |H |W |G |                    |G |
       |G |E |D |E |A |O |                    |V |
       |L |R |L |D |K |B |                    |A |
       |E |E |K |M |E |D |                    |L |
       |  |Q |  |S |  |R |                    |I |
       |  |  |G |  |T |  |                    |D |
         |---------------------------------------------|
|RB-X22|   MAKE PRESENT DEFERRED QUEUE (PCB INDEX)   |MPQLINK
         |---------------------------------------------|
|RB-X21|   RELEASE PAGE COUNT                        |PAGECNT
         |---------------------------------------------|
|RB-X20|   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-X17|     SUBREGION ASSIGNMENT STATE            | I  |SAS
|      |C |R |R |R |R |                            | O  |
|      |A |E |O |E |E |                            | S  |
|      |C |F |C |F |F |                            | T  |
|      |H |1 |  |2 |3 |                            |    |
          |----------------------------------------------|
|RB-X16|     SUBREGION SIZE IN PAGES                     |SS
          |--|-------------------------------------------|
|RB-X15|V|   SUBREGION DISPLACEMENT IN MAIN MEM. PAGES   |SD
          |----------------------------------------------|
|RB-X14|     WRITE REQUEST POINTER                       |WREQP
          |----------------------------------------------|
|RB-X13|     OBJECT                                      |OBJIDENT
|      |-                                               -|
|      |     IDENTIFIER                                  |
          |----------------------------------------------|
|RB-X11|     FREEZE COUNT      |    LOCK COUNT            |LKFZCNT
          |----------------------|-----------------------|
|RB-X10|     WRITE DISABLE      |    I/O FROZEN           |WDIOFZCNT
|      |       COUNT           |      COUNT              |
          |----------------------|-----------------------|
| RB-X7|     LDEV              |    HIGH ORDER DISC       |HODA
|      |                       |      ADDRESS            |
          |----------------------|-----------------------|
| RB-X6|     LOW ORDER DISC ADDRESS                      |LODA
          |----------------------------------------------|
| RB-X5|                                                 |
          |----------------------------------------------|
| RB-X4|                                                 |
          |----------------------------------------------|
| RB-X3|     TIME OF                                     |ARRTIME
|      |-                                               -|
|      |     ARRIVAL                                     |
          |----------------------------------------------|
| RB-X1|                                                 |
          |----------------------------------------------|
```

Global Region Header (Assigned Regions)

```
           0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
          |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
|RB-X30|     REGION ASSIGNMENT STATE                     |RAS
|      |R |R |A |C |C |L |F |I |L |                 |B  |
|      |S |E |V |L |C |K |Z |O |S |                 |L  |
|      |S |S |  |N |  |P |W |F |T |                 |M  |
|      |  |  |  |D |  |  |  |Z |T |                 |I  |
|      |  |  |  |  |  |  |  |N |  |                 |P  |
          |----------------------------------------------|
|RB-X27|     REGION SIZE IN PAGES                        |RS
          |----------------------------------------------|
|RB-X26|                                                 |
          |----------------------------------------------|
|RB-X25|                                                 |
          |----------------------------------------------|
|RB-X24|                                                 |
          |----------------------------------------------|
|RB-X23|                                                 |
          |----------------------------------------------|
|RB-X22|                                                 |
          |----------------------------------------------|
|RB-X21|                                                 |
          |----------------------------------------------|
|RB-X20|                                                 |
          |----------------------------------------------|
```

Subregion Header (Assigned Regions)

```
           0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
          |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
|RB-X17|     SUBREGION ASSIGNMENT STATE            | I  |SAS
|      |C |R |R |R |R |                            | O  |
|      |A |E |O |E |E |                            | S  |
|      |C |F |C |F |F |                            | T  |
|      |H |1 |  |2 |3 |                            |    |
          |----------------------------------------------|
|RB-X16|     SUBREGION SIZE IN PAGES                     |SS
          |--|-------------------------------------------|
|RB-X15|V|   SUBREGION DISPLACEMENT IN MAIN MEM. PAGES   |SD
          |--|-------------------------------------------|
|RB-X14|     WRITE REQUEST POINTER                       |WREQP
          |----------------------------------------------|
|RB-X13|     OBJECT                                      |OBJIDENT
|      |-                                               -|
|      |     IDENTIFIER                                  |
          |----------------------------------------------|
|RB-X11|     FREEZE COUNT      |    LOCK COUNT            |LKFZCNT
          |----------------------|-----------------------|
|RB-X10|     WRITE DISABLE      |    I/O FROZEN           |WDIOFZCNT
|      |       COUNT           |      COUNT              |
          |----------------------|-----------------------|
| RB-X7|     LDEV              |    HIGH ORDER DISC       |HODA
|      |                       |      ADDRESS            |
          |----------------------|-----------------------|
| RB-X6|     LOW ORDER DISC ADDRESS                      |LODA
          |----------------------------------------------|
| RB-X5|                                                 |
          |----------------------------------------------|
| R9-X4|                                                 |
          |----------------------------------------------|
| RB-X3|     TIME OF                                     |ARRTIME
|      |-                                               -|
|      |     ARRIVAL                                     |
          |----------------------------------------------|
| RB-X1|                                                 |
          |----------------------------------------------|
```

Subregion Header (Cached Regions)

```
           0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
          |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
|RB-X17|     SUBREGION ASSIGNMENT STATE            | I  |SAS
|      |C |R |R |R |R |                            | O  |
|      |A |E |O |E |E |                            | S  |
|      |C |F |C |F |F |                            | T  |
|      |H |1 |  |2 |3 |                            |    |
          |----------------------------------------------|
|RB-X16|     SUBREGION SIZE IN PAGES                     |SS
          |--|-------------------------------------------|
|RB-X15|V|   SUBREGION DISPLACEMENT IN MAIN MEM. PAGES   |SD
          |----------------------------------------------|
|RB-X14|     WRITE REQUEST POINTER                       |WREQP
          |----------------------------------------------|
|RB-X13|     OBJECT                                      |OBJIDENT
|      |-                                               -|
|      |     IDENTIFIER                                  |
          |----------------------------------------------|
|RB-X11|     PREVIOUS CACHED REGION (ADDRESS OF PD       |PD
|      |-                                               -|
|      |       FIELD OF PREVIOUS CACHED REGION)          |
          |----------------------------------------------|
| RB-X7|     LDEV              |    HIGH ORDER DISC       |HODA
|      |                       |      ADDRESS            |
          |----------------------|-----------------------|
| RB-X6|     LOW ORDER DISC ADDRESS                      |LODA
          |----------------------------------------------|
| RB-X5|     NEXT CACHED REGION (ADDRESS OF ND           |ND
|      |-                                               -|
|      |       FIELD OF NEXT CACHED REGION)              |
          |----------------------------------------------|
| RB-X3|     TIME OF                                     |ARRTIME
|      |-                                               -|
|      |     ARRIVAL                                     |
          |----------------------------------------------|
| RB-X1|     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, Region Map Flag
.(9:6) Not used
.(15:1) Blocked Lock Migration in Progress Flag

IOCNT, On-Going I/O Count
= # of on-going I/Os 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
.(15:1) Message Valid Flag

INITINFO, Initiation Message Auxiliary Information
= DRQ relative index of segment read disc request if
INITMSG.QREADREQ =1
or
QREADREQ = +/- 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
.(3:1) Reference 2
.(4:1) Reference 3
.(13:3) I/O Status from region fetch

SS, Subregion Size (in pages)

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 cached domain which
is mapped (i.e., OBJIDENT = 30000/non zero number)
this word is non zero. If the cached 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, I/O freeze count
.(0:8) Not used
.(8:8) Number of times region has been iofrozen

For regions belonging to cached domains, the above two words
contain the absolute address of the PD field in the previous
region belonging to a cached 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 cached domain link for cached domain regions
only. Contains the absolute address of the ND field
of the next cached 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 cached 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 cached
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 X250
through X253 contain the respective head and tail (bank & address) of the
available region list. These four words have in essence replaced the ARSBN
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 X250 & X251 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 #
 X  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|      CODE FOR                           |4  \
     |-                                      -|    |
     5|-        INITIAL PROGRAMS               |5  |
     |-                                      -|    |
     6|-        "BOOTSTRAP"                    |6  |
     |-                                      -|    |
     7|-        SEGMENT                        |7  |
     |-                                      -|    |
    10|-                                      -|8  |
     |-                                      -|    |
    11|-                                      -|9  |
     |-                                      -|    > 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 #
   X    .                                            .
        |                                            |
     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
        |--------------------------------------------|
        .                                            .

SYSDB   |--------------------------------------------|
----->  |                                            |        ---> NOTE: INITIAL
X130/131|                                            |             TRIES TO
        |            SYSTEM DIRECTORY                |             ALLOCATE
        |                                            |             DIRECTLY AFTER
        |--------------------------------------------|             THE FREE SPACE
        |                                            |             MAP. HOWEVER,
        |                                            |             THIS MAY
        |          VIRTUAL MEMORY AREA               |             VARY DEPENDING
        |                                            |             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 |  0                                  |0       WORDS 0-5 CONTAIN
      |-----------------------------------|         THE ASCII STRING
    1 |  0                                  |1       "SYSTEM DISC " FOR
      |-----------------------------------|         THE SYSTEM DISC,
    2 |  0                                  |2       ONLY.
      |-----------------------------------|
    3 |  0                                  |3
      |-----------------------------------|
    4 |  0                                  |4
      |-----------------------------------|
    5 |  0                                  |5
      |-----------------------------------|
    6 |         DISC TYPE  |DISCSUBTYPE     |6
      |-----------------------------------|
    7 |   ROLLBACK COLD LOAD ID **SEE NOTE BELOW** |7
      |-----------------------------------|
   10 |   "3"             |   "0"          |8
      |-----------------------------------|
   11 |   "0"             |   "0"          |9       IF WORD X11
      |-----------------------------------|         CONTAINS A "1"
   12 |                                     |10      A FORMER SYSTEM
      |-----------------------------------|         VOLUME HAS BEEN
   13 |                                     |11      SCRATCHED.
      |       VOLUME NAME                  |
   14 |                                     |12
      |-----------------------------------|
   15 |                                     |13
      |-----------------------------------|
   16 |       UNUSED                        |14
      |-----------------------------------|
   17 |       UNUSED                        |15
      |-----------------------------------|
   20 |   VOLUME SET ID   **SEE NOTE BELOW**|16
      |-----------------------------------|
 21-24|  UNUSED [words 17-20 (X21-X24) UNUSED]|17-20
      |-----------------------------------|
   25 |  SYSUCS64.PUB.SYS High Order Disc Address |21  *6X/70 UCS
      |-----------------------------------|             IMAGE
   26 |  SYSUCS64.PUB.SYS Low Order Disc Address  |22  POINTER
      |-----------------------------------|
   27 |  SYSUCS37.PUB.SYS High Order Disc Address |23  *37 UCS
      |-----------------------------------|             IMAGE
   30 |  SYSUCS37.PUB.SYS Low Order Disc Address  |24  POINTER
      |-----------------------------------|
```

System Volume (Cont.)

```
   31 |  UCSLE1.PUB.SYS High Order Disc Address |25  *MICRO/XE
      |-----------------------------------|             UCS IMAGE
   32 |  UCSLE1.PUB.SYS Low Order Disc Address  |26  POINTER
      |-----------------------------------|
   33 |  UCSLE2.PUB.SYS High Order Disc Address |27  *MICRO/LX/GX
      |-----------------------------------|             UCS IMAGE
   34 |  UCSLE2.PUB.SYS Low Order Disc Address  |28  POINTER
      |-----------------------------------|
   35 |                                     |29
      |-----------------------------------|
      |       RESERVED                      |
      |                                     |
      |-----------------------------------|
  170 |                                     |120
      |-----------------------------------|
  171 |   DISC FREE SPACE MAP OK FLAG       |121
      |-----------------------------------|
  172 |   DISC FREE SPACE MAP DESCRIPTOR    |122
      |     TABLE CHECKSUM                  |
      |-----------------------------------|
  173 |   DISC FREE SPACE DESCRIPTOR TABLE  |123
      |     DIRTY FLAG                      |
      |-----------------------------------|
  174 |-  DISC FREE SPACE DESCRIPTOR        -|124
  175 |     TABLE ADDRESS                   |125
      |-----------------------------------|
  176 |-  DISC FREE SPACE BITMAP ADDRESS    -|126
  177 |                                     |127
      |-----------------------------------|
```

* UCS image pointers point to the start of the UCS
  data (File label address + 1). Always on LDEV 1.

** As of V-Delta-5 (G.03.05) the way the COLDLOAD ID's are
  used has been changed. A Volume set ID (VID) has been
  created to logically link together a set of discs. Originally
  the COLDLOAD ID peformed this, as well as enabling the FILE
  SYSTEM to tell if the file was open when the system failed.
  The VID is NOT changed on each system start, only on a RELOAD.
  In order to maintain backward compatibility, the old COLDLOAD
  ID locations have been changed to ROLLBACK COLDLOAD ID's. The
  actual COLDLOAD ID (for FILESYS) is in the Disc Cold Load info
  Table (DCLT) word 46 (X56).

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    |MEDIA 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 |13 /
      |                   |   NUMBER       |
      |-----------------------------------|
   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
   22 |     END OF TAPE (EOT)               |18  | INFO
      |-----------------------------------|   |
   23 |-  DOUBLE ADDRESS OF                -|19  |
   24 |     END OF DATA (EOD)               |20 /
      |-----------------------------------|
   25 |  SYS_CS64.PUB.SYS High Order Disc Address |21  ICF UCS
      |-----------------------------------|             IMAGE
   26 |  SYSUCS64.PUB.SYS Low Order Disc Address  |22  POINTER
      |-----------------------------------|
```

Serial Volume (Cont.)

```
     |-------------------------------------------------|
|27| |      (See SYSTEM VOLUME words X25-X34)          |23
 ~   |                                                 |  ~
     |           RESERVED FOR FUTURE WCS               |
|122||                                                 |82
     |-------------------------------------------------|
|123||                   CYL                           |83
     |-------------------------------------------------|
|124||      HEAD           |        SECTOR             |84
     |-------------------------------------------------|
```

* MEDIA TYPE is the device subtype for all serial volumes except cartridge
  tape. For cartridge tape, this field is always 0 (the HP 9110 subtype),
  despite a different actual cartridge tape subtype. This allows both
  forward and backward interchangeability of cartridges between the HP 9110
  and HP 9144.

Master Volume

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

Master Volume (Cont.)

```
                      0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
                     |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
                  25||                                               |21
                  26||             GROUP                             |22
                  27||             NAME                              |23
                  30||                                               |24
                     |-----------------------------------------------|
                  31||                                               |25
                  32||           VOLUME SET                          |26
                  33||             NAME                              |27   HEADER
                  34||                                               |28
                     |-----------------------------------------------|
VS VTAB           35||                                               |29
HEADER +             |-----------------------------------------------|
8 ENTRIES         36||VCOUNT         |             VMASK             |30
COPIED FROM          |-----------------------------------------------|
VSET DEFN         37||                                               |31
IN SYSTEM         40||           VOLUME                              |32
DIRECTORY         41||            NAME                               |33   VOLUME
                  42||                                               |34   ENTRY
                     |-----------------------------------------------|    .
                  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                |122
                     |           TABLE CHECKSUM                      |
                     |-----------------------------------------------|
                 173||     DISC FREE SPACE DESCRIPTOR TABLE          |123
                     |           DIRTY FLAG                          |
                     |-----------------------------------------------|
                 174||     DISC FREE SPACE DESCRIPTOR                |124
                     |-                                             -|
                 175||           TABLE ADDRESS                       |125
                     |-----------------------------------------------|
                 176||                                               |126
                     |-     DISC FREE SPACE BITMAP ADDRESS          -|
                 177||                                               |127
                     |-----------------------------------------------|
```

Slave Volume

```
                      0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
                     |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
                   0||                                               |0
                   1||                                               |1
                   2||                0                              |2
                   3||                                               |3
                   4||                                               |4
SC = SCRATCH       5||                                               |5
    VOLUME           |--|--|--|-------------------------|------------|
MV = MASTER        6||SC|MV|SR|        TYPE             |  SUB-TYPE  |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                |122
                     |           TABLE CHECKSUM                      |
                     |-----------------------------------------------|
                 173||     DISC FREE SPACE DESCRIPTOR TABLE          |123
                     |           DIRTY FLAG                          |
                     |-----------------------------------------------|
                 174||     DISC FREE SPACE DESCRIPTOR                |124
                     |-                                             -|
                 175||           TABLE ADDRESS                       |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
    |                .                   |
    -                .                   -
    |                .                   |
170|     DEFECTIVE TRACK NUMBER          |DTC |120
    |-----------------------------------|----|
171|                                            |121
172|     RESERVED FOR                           |122
173|     FUTURE USE                             |123
174|                                            |124
    |-----------------------------------|
175|     CHECKSUM                               |125 DTTCKSUM
    |-----------------------------------|
176|     NEXT AVAILABLE ALTERNATE TRACK         |126
    |-----------------------------------|
177|     LOGICAL DISC PACK SIZE (CYLINDERS)     |127
    |-----------------------------------|
         OR # OF TRACKS IF FM 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.

> DTTCKSUM (System Volumes only): This is an EXCLUSIVE-OR
> checksum(from a base of -1) of the DTT excluding word X175.
> Each time a suspect track is inserted or modified, a new
> checksum is calculated and stored in word X175. At system
> startup INITIAL recalculates the checksum and compares against
> original value at X175. If the checksums do not match a
> COLDLOAD ERROR 202 (MOUNT CORRECT VOLUMES OR RELOAD) will occur.

Defective Sector Table (DSCT -- Sector 1 of Disc)
(The DSCT Exists On Device Type 3 (CS-80) Discs, Except Cartridge Tape)

```
       0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
      |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
    0|    NUMBER OF ENTRIES IN THE TABLE         |0
      |--------------------------------------|
   X1|    INDEX TO THE FIRST ENTRY (6)           |1
      |--------------------------------------|
   X2|    ENTRY SIZE (2)                         |2
      |--------------------------------------|
   X3|    MAXIMUM NUMBER OF ENTRIES (X75)        |3
      |--------------------------------------|
   X4|    0 (RESERVED)                           |4
      |--------------------------------------|
   X5|    0 (RESERVED)                           |5
      |--------------------------------------|
   X6|    FIRST DEFECTIVE SECTOR ENTRY           |6
   X7|        (DOUBLE-WORD LOGICAL SECTOR ADDRESS)|7
      |--------------------------------------|
  X10|    SECOND ENTRY                           |8
  X11|                                           |9
      |--------------------------------------|
  X12|    THIRD ENTRY                            |10
  X13|                                           |11
      |                 .                     |
      -                 .                     -
      |                 .                     |
X176|    MAXIMUM DEFECTIVE SECTOR ENTRY         |126
X177|                                           |127
      |--------------------------------------|
```

Unlike the DTT, entries in the DSCT are not permanent. Once a suspect sector
is handled by INITIAL, SDISC, 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.

```
      |--------------------------------------|
  X0|                                            |0
      |     RESERVED AREA                      .|
      |     BIT MAP                             |
      |                 .                      |
      |                 .                      |
 X30|                                           |24
      |--------------------------------------|
 X31|                                           |25
      |     RESERVED FOR                        |
      |     FUTURE USE                          |
      |                 .                      |
      |                 .                      |
X177|                                           |127
      |--------------------------------------|
```

Disc Cold Load Information Table (Sectors X34-X36)

	0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15	
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
10	INITIAL'S S VALUE	INITS
11	SYSDISC TYPE / SUBTYPE	DISCTST
12	ROLLBACK COLDLOAD ID **See note below**	COLD'LOAD'ID'
13	LOG FILE NUMBER	LOG'FILE'NUM'
14	DIRECTORY DISC	DIRADR
15	ADDRESS	
16	LDEV 1 VIRTUAL MEMORY	VIRMEMADDR
17	DISC ADDRESS	
20	# LOG PROCS	NLOGPROCS
21	LOG ID'S	LOGIDS
22	RIN TABLE	RINADR
23	DISC ADDRESS	
24	DIRECTORY SIZE	DIRSECT
25	#SECTORS IN VIRTUAL MEMORY REGION OF LDEV 1	SECTORS IN LDEV1VM VIRMEMSECT
26	VOLUME SET ID (VID) **See note below**	
27	RIN TABLE SIZE	RINSECT

Disc Cold Load Information Table (Cont.)

```
     0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
     |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
30|  # OF RINS                             |RINS
31|  # OF GLOBAL RINS                      |GRINS
     |                          |--|--|--| | TL=TAPE COLD LOAD
32|                          |TL|RL|RY| LOAD MODE
     |                          |--|--|--| | RL=RELOAD
     |                                     | RY=RECOVERY
33|  MAX VOL          |  HIGH VOL          |H'VOL'
34|  DISC COLD LOAD ENTRY POINT            |DISCENTRY
35|  SYSTEM DISC DRT NUMBER                |SYSDISCDRT
36|  JOB MASTER TABLE                      |JMATLOC
37|      DISC ADDRESS
40|
    IDD DISC ADDRESS                       |IDDLOC
41|
42|
    ODD DISC ADDRESS                       |ODDLOC
43|
44|  WELCOME MESSAGE (DST X57)
                                           |LOGONLOC1
45|      DISC ADDRESS
46|  WELCOME MESSAGE (DST X60)
                                           |LOGONLOC2
47|      DISC ADDRESS
50|
    LOG ID ADDRESS
51|
52|
    LOG TAB ADDRESS
53|
54|  LOG ID SIZE
55|  LOG TAB SIZE
56|  COLDLOAD ID      **See note below**
```

G.23.00
3- 14

Disc Cold Load Information Table (Cont.)

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

G.23.00
3- 15

Disc Cold Load Information Table (Cont.)

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

G.23.00
3- 16

Disc Cold Load Information Table (Cont.)

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

G.23.00
3- 17

Disc Cold Load Information Table (Cont.)

```
|----------------------------------|
|          SEGMENT SIZE            |FREFTR+75
|----------------------------------|
|                          DEFDATA |
|        MEMORY ADDRESS     TABLE   |
|                          LOOKUP   |
|------------------------- BUFFER  |
|                                  |
|         DISC ADDRESS             |
|                                  |
|----------------------------------|FREFTR+80
|              .                   |
|              .                   |
|      (INITIAL'S SEGMENTS)        |
|             ININ                 |
|              .                   |
|              .                   |
|----------------------------------|
```

INITIAL Program CST Map

LOGICAL CST#	PHYSICAL 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

** As of V-Delta-5 (G.03.05) the way the COLDLOAD ID's are
used has been changed. A Volume set ID (VID) has been
created to logically link together a set of discs. Originally
the COLDLOAD ID peformed this, as well as enabling the FILE
SYSTEM to tell if the file was open when the system failed.
The VID is NOT changed on each system start, only on a RELOAD.
In order to maintain backward compatibility, the old COLDLOAD
ID locations have been changed to ROLLBACK COLDLOAD ID's. The
actual COLDLOAD ID (for FILESYS) is in the Disc Cold Load info
Table (DCLT) word 46 (X56). This value is put in SYSGLOB(X75).

SYSDUMP/Initial Communication Record (Sector X37)

```
                 1 1 1 1 1 1
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
     |-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|
|  0| MIT VERSION                    |0
     |-------------------------------|
|  1| MIT UPDATE                     |1
     |-------------------------------|
|  2| MIT FIX                        |2
     |-------------------------------|
|  3| VERSION                        |3
     |-------------------------------|
|  4| UPDATE                         |4
     |-------------------------------|
|  5| FIX                            |5
     |-------------------------------|
|  6| EXP. SOFTWARE SYSTEM NO.       |6
     |-------------------------------|
|  7| HIGHEST DPT                    |7
     |-------------------------------|
| 10| HIGHEST LDEV                   |8
     |---------------|---------------|
| 11| MAX VOL       | HIGH VOL       |9
     |---------------|---------------|
| 12| # OF ADD'L DRIVERS             |10
     |-------------------------------|
| 13| COLD LOAD COUNT                |11
     |-------------------------------|
| 14| FILES DUMPED                   |12
     |-----------------------|-|-|-|
| 15| SERIAL DISC LOAD      |F|D|S|13
     |-----------------------|-|-|-|
| 16| TAPE RECORD SIZE              |14
     |-------------------------------|
| 17| DISC COLD LOAD ENTRY          |15
     |-------------------------------|
| 20| MAX INITIAL SEG SIZE          |16
     |-------------------------------|
| 21| SPARE                         |17
     |-------------------------------|
| 22| SPARE                         |18
     |-------------------------------|
| 23| SPARE                         |19
     |-------------------------------|
| 24| DEV CLASS TAB SIZE            |20
     |-------------------------------|
| 25| TERM DESCRIPTOR SIZE          |21
     |-------------------------------|
| 26| OLD VTAP SIZE                 |22
     |-------------------------------|
| 27| OLD INFO SIZE                 |23
     |-------------------------------|
```

F=>Set if FOS Sysdump
D=>Set if Future Date Sysdump
S=>Set if Serial Disc Sysdump

SYSDUMP/Initial Communication Record (Cont.)

```
     |-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|
| 30| CS TABLE SIZE                 |24
     |-------------------------------|
| 31| TABLE LOOKUP BUF SIZE         |25
     |-------------------------------|
| 32| TABLE LOOKUP BUF ENTRIES      |26
     |-------------------------------|
| 33| SYSTEM TAPE LDEV #            |27
     |-------------------------------|
| 34| SPARE                         |28
     |-------------------------------|
| 35| SPARE                         |29
     |-------------------------------|
| 36| CONVERSION BITS WORD 1    |M|30
     |--------------------------|-|
| 37| MIT FIX LEVEL INDICATOR ** |31
     |-------------------------------|
| 40| CONVERSION BITS WORD 3        |32
     |-------------------------------|
| 41| CONVERSION BITS WORD 4        |33
     |-------------------------------|
| 42| SPARE                         |34
     |-------------------------------|
| 43| SPARE                         |35
     |-------------------------------|
| 44| SPARE                         |36
     |-------------------------------|
| 45| SPARE                         |37
     |-------------------------------|
| 46| SPARE                         |38
     |-------------------------------|
| 47| SPARE                         |39
     |-------------------------------|
| 50| LOG FILE NUMBER               |40
     |-------------------------------|
| 51| LAST FULLBACKUP DUMP DATE     |41
     |-------------------------------|
```

M=(15:1) MPE Version
0=MPE (G.00.00)
1=MPE (G.01.00)

** As of V-Delta-5 (G.03.05) word 31 (X37) of the SYSDUMP/INITIAL
COMMUNICATION RECORD is now used as the MIT FIX LEVEL INDICATOR.
It was previously used as the CONVERSION BITS WORD 2.

As of V-Delta-5 it (word X37) will contain the value X170005.
The '5' in bits (13:3) is added by INITIAL during the update
to V-Delta-5 so INITIAL will know that it is using the new
COLDLOAD ID/VID mechanism.
Bits (0:4) signal that the following tables have been converted
to MPE V/E format:
                    0 - IO tables converted.
                    1 - Cold Load Info table converted.
                    2 - Rin table converted.

3 - Sysdump initial Communication record.

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:

Sector %40

```
|---------------------------------------------|
  0|                                           |0
   |-                                         -|
  1|                                           |1
   |-                                         -|
  2|       RESERVED FOR FUTURE SYSTEM USE      |2
  -                                            -
   |-                                         -|
 24|                                           |20
  |---------------------------------------------|
 25|       SYSTEM LOGGING FILE NUMBER          |21
  |---------------------------------------------|
 26|   NETWORK MANAGEMENT LOGGING FILE NUMBER  |22
  |---------------------------------------------|
 27|   NETWORK MANAGEMENT TRACE FILE NUMBER    |23
  |---------------------------------------------|
 30|     FULL/PARTIAL COMMAND DUMP DATE        |24
  |---------------------------------------------|
 31|                                           |25
  |---------------------------------------------|
 32|                                           |26
   |-                                         -|
 33|       NOT CURRENTLY ASSIGNED              |27
   |-                                         -|
 34|                                           |28
   |-                                         -|
  -                                            -
   |-                                         -|
377|                                           |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        |
             |---------------------|
```
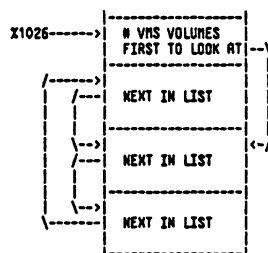
VDSMTAB Entry 0 Format

```
             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           |VMSVOLUMECNT
         |   VIRTUAL MEMORY                        |
             |----------------------------------------|
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 |** See below
             |----------------------------------------|
VDSMTAB07|   LEAST # OF VM PAGES THAT HAVE         |
         |   EVER BEEN AVAILABLE                   |
             |----------------------------------------|
         |                                        |
             -                                      -
         |                                        |
         |   VDSMTAB %10-%17 UNASSIGNED            |
         |                                        |
             |----------------------------------------|
```

** This 16 bit field can only accommodate 32K Pages or 255K sectors. Each volume can have up to 255K sectors of virtual memory. This word will overflow if there are more than 255K total VM pages configured on all system discs. MPE does not use this word. It instead uses the general VDSMTAB entry for each volume to find out the total virtual memory sectors on a particular volume.

VDSMTAB General Entry Format

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

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

Volume Table

SIR #22=X26
DST #29=X35

Zero Entry

```
OCTAL  0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 DECIMAL
WORD  |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--| WORD
    0| # OF ENTRIES (NOT  |                        | 0
     |   COUNTING        | ENTRY SIZE=X16         |
     |   ZERO)           |                        |
     |------------------------------------------- |
    1| ROLLBACK COLDLOAD ID  **See note below**   | 1
     |------------------------------------------- |
    2| NUMBER OF VOLUMES                           | 2
     |------------------------------------------- |
    3| ROLLBACK VIRTUAL MEMORY INTEGRITY NUMBER    | 3
     |------------------------------------------- |
    4| VOLUME SET ID      **See note below**       | 4
     |------------------------------------------- |
    5| VIRTUAL MEMORY INTEGRITY NUMBER  **         |
     |------------------------------------------- |
     -                                            -
   15|                                            | 13
     |------------------------------------------- |
```

** As of V-Delta-5 (G.03.05) the way the COLDLOAD ID's are
   used has been changed. A Volume set ID (VID) has been
   created to logically link together a set of discs. Originally
   the COLDLOAD ID peformed this, as well as enabling the FILE
   SYSTEM to tell if the file was open when the system failed.
   The VID is NOT changed on each system start, only on a RELOAD.
   In order to maintain backward compatibility, the old COLDLOAD
   ID locations have been changed to ROLLBACK COLDLOAD ID's. The
   actual COLDLOAD ID (for FILESYS) is in the Disc Cold Load info
   Table (DCLT) word 46 (X56).

Typical Private Volume Entry

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

Typical System Volume Entry

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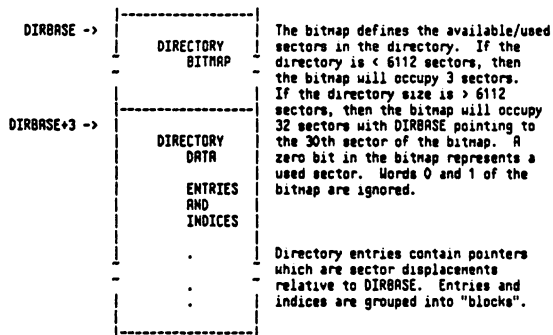
CHAPTER 4 DIRECTORY

Introduction to the Directory

SYSGLOB cells:

    DIRBASE <----absolute disc addr of base [SYSGLOB+Z130 AND Z131]

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
DIRBASE+3 ->|                  |      sectors, then the bitmap will occupy
            |    DIRECTORY     |      32 sectors with DIRBASE pointing to
            |    DATA          |      the 30th sector of the bitmap.  A
            |                  |      zero bit in the bitmap represents a
            |    ENTRIES       |      used sector.  Words 0 and 1 of the
            |    AND           |      bitmap are ignored.
            |    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)
* SYSREBSIZE   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.
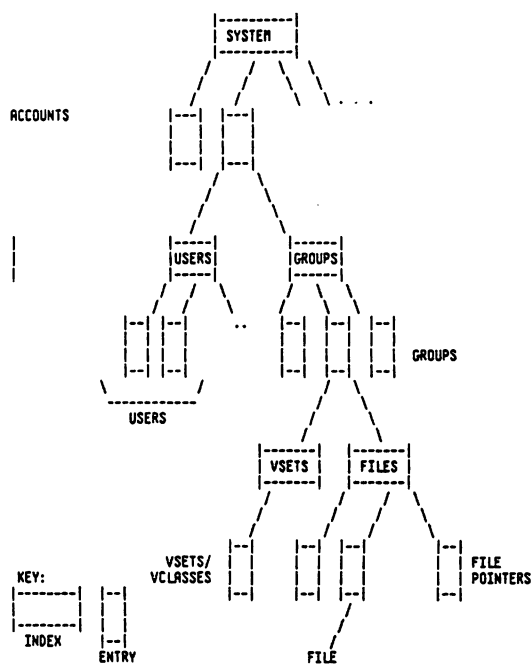
Directory Data Segment

```
   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  |131
    |    INDEX PRT)    |
    |------------------|
 204|  XANAME (DB REL ADDR)|132
    |------------------|
 205|  XGUNAME (DB REL ADDR)|133
    |------------------|
 206|  XFNAME (DB REL ADDR)|134
    |------------------|
 207|  XASEC (ACCOUNT  |135
    |    SECURITY)     |
    |------------------|
 210|                  |136
    -  XGSEC (GROUP    -
 211|     SECURITY)    |137
    |------------------|
 212|  SIRRETURN (FROM |138
    |    GETSIR)       |
    |------------------|
213-240  DIRECTORY POINTER  139-160 \
    |        "A"        |             \
    |------------------|              > See Directory
241-266  DIRECTORY POINTER  161-182  / Pointer Area
    |        "B"        |             /
    |------------------|
 267|  SYS.ACCT.INDEX  |183
    |    BLOCK SIZE    |
    |------------------|
 270|  LDEV  |   PV    |184
    |------------------|
 271|  DIRECTORY ADDRESS|185
    |------------------|
 272|  PRIVATE VOLUME  |186
    |    DIRECTORY SIZE|
    |------------------|
 273|                  |187
    |------------------|
 274|                  |188
    |------------------|
```

```
 275|------------------|189
    |------------------|
 276|                  |190
    |------------------|
 277|                  |191
    |------------------|
 300|                  |192
    |------------------|
 301|                  |193
    |------------------|
 302|                  |194
    |------------------|
 303|                  |195
    |------------------|
 304|                  |196
    |------------------|
 305|                  |197
    |------------------|
 306|  DISTRIBUTION    |198
    -                  - GOODPERCENT=.85
 307|  FACTOR          |199
    |------------------|
 310|  BASE            |200
    |------------------|
 311|                  |201
    |  DA AREA         |
    -                  -DDSBUSIZE
    |                  |
    |------------------|
    |                  |
    |  WORK AREA       |
    -  (SIZE OF        -MAX
    |   LARGEST        |
    |   ENTRY)         |
    -                  -
    |------------------|
1145|                  |613
    |  DB AREA         -DDSBUSIZE
    -                  |
    |------------------|
```

Directory Pointer Area [DA or DB] DST=X24 SIR=8

```
            1 1 1 1 1 1
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
^  -|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-
|  |    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
|  |------------------------------|
*  |  # WORDS USED IN BLOCK       |147/169 USED
|  |------------------------------|
|  |  BLOCK SIZE (SECTORS)        |148/170 BSIZE
|  |------------------------------|
|  |  BLOCK SIZE (WORDS)          |149/171 BWSIZE
|  |------------------------------|
|  |  MAX # ELEMENTS/BLOCK        |150/172 BFACTOR
|  |-|-|-|-|-|------------|-------|
|  |I|P| TY |ELEMENT SIZE |BL SIZE|151/173 MISCWD
|  | | |    |  (WORDS)    |(SECTs)|
|  |-|-|-|-|-|------------|-------|
|  |  NUMBER OF ELEMENTS          |152/174 XCOUNT
|  |------------------------------|
|  |  NUMBER OF ACCESSORS         |153/175 PCOUNT
|  |------------------------------|
|  |  ENTRY TOTAL                 |154/176 ETOTAL
|  |-|-|-|-|-|------------|-------|
|  |O|P| TY | ENTRY SIZE  |BL SIZE|155/177 EMISCWD
|  | | |    |  (WORDS)    |(SECTs)|
|  |-|-|-|-|-|------------|-------|
|  |  FATHER INDEX POINTER        |156/178 PINDEXP
|  |---------------|--------------|
** |   FATHER      |   FATHER     |157/179
|  |-      -|-     -|
|  |   NAME     |-|   NAME        |158/180 PNAME    TY = 0-FILE
|  |-      -|-     -|                            1-GROUP
|  |           |-|                |159/181                2-ACCT
|  |-      -|-     -|                            3-USER
|  |           |-|                |160/182                4-VSD
v  |---------------|--------------|                I = 0-ENTRY BLOCK
* Indexes Only                                       1-INDEX BLOCK
** Indexes and Entries                            P = PURGE FLAG
```

Directory Space Data Segment (DIRSDS)

```
    DST=21 (X25)
    SIR=8 (X10)

             1 1 1 1 1 1
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
   -|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-
  0| LOGICAL DEV |   BIT MAP      |
   |---------------|
  1|   BASE SECTOR ADDRESS        |DS'BASE
   |------------------------------|
  2|  POINTER TO LAST WORD        |DS'LAST'WORD
   |      IN BUFFER               |
   |------------------------------|
  3|  POINTER TO FIRST WORD       |DS'FIRST'WORD
   |      IN BUFFER               |
   |------------------------------|
  4|  SIZE OF DIRECTORY           |DS'DIR'SIZE
   |     IN SECTORS               |
   |-|-|-|-|----------------------|
  5|D|E|S|P|                      |DS'FLAGS
   |-|-|-|-|----------------------|
  6|  FIRST CURRENT SECTOR        |DS'CUR'SECTOR
   |      IN BUFFER               |
   |------------------------------|
  7|  DISC ADDRESS OF CURRENT     |
   |     PART OF BIT MAP          |DS'ADDR
 10|     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 PTR TO LAST    |DS'SYS'LAST
   |------------------------------|
 15|  SYSTEM SAVED PTR TO FIRST   |DS'SYS'FIRST
   |------------------------------|
 16|  SYSTEM SAVED CURRENT        |DS'SYS'CUR
   |      SECTOR                  |
   |------------------------------|
 17|  SAVED DIRECTORY SIZE        |DS'SYS'SIZE
   |------------------------------|
 20|  LDEV THAT LAST ERROR        |DS'ERROR'LDEV
   |      OCCURRED                |
   |------------------------------|
```

Directory Space Data Segment (DIRSDS) (Cont.)

```
     |------------------------------|
 21| TYPE OF ERROR THAT             |DS'ERROR'TYPE
     |      OCCURRED                |
     |------------------------------|

     |------------------------------|
     | 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'BASE0.(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 X577 + DS'HEADER. If there exists 3 full sectors in the buffer, then it will have the value X600 + DS'HEADER - 1 or X621. 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 X600. Legal values are X200, X400, and X600.

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
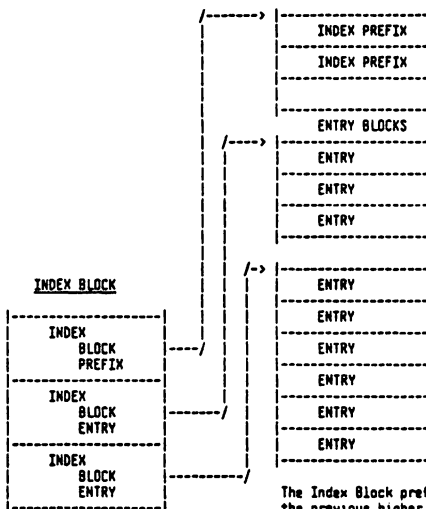

The Index Block prefix points back to the previous higher level. The Index Block entries point to the entry blocks.

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.
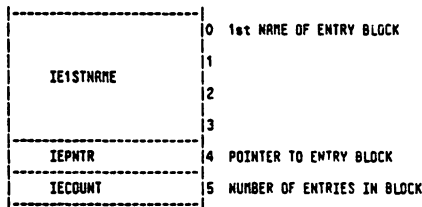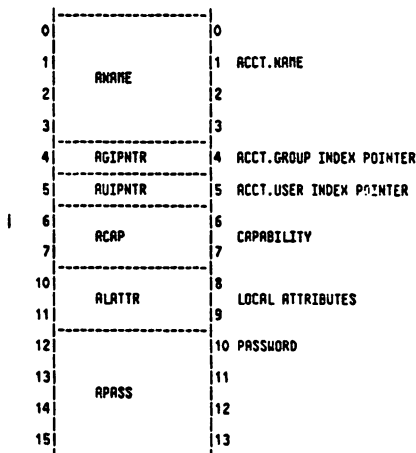
Index Block Prefix (10 Words)
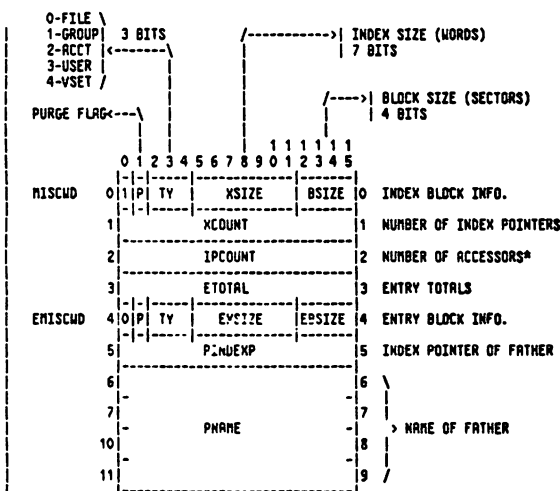
*The count is incremented by each access that uses and relies upon a pointer to the index block, i.e., it is guaranteed not to be purged while the count is not = 0.

Index Entry (6 Words)

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

Account Entry (X36 Words)

Account Entry (Cont.)

```
16|-----------------|14 DISC FILE SPACE COUNT (SECTORS)
  | RDFSCOUNT       |
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|E|U|      |   27 MAX.JOB PRIORITY
|  |-|-|-|-|-|-|-|-|
| 34| COMM FILE REC |   28 COMMAND FILE LOCATION OF
|  |   # ACCT      |      ACCOUNT UDCS          HARD CODED
|  |---------------|                            1  0
| 35| COMM FILE REC |   29 COMMAND FILE LOCATION OF
|  |   # SYS       |      SYSTEM UDCS (SYS ACCT ONLY)
|  |---------------|
```

```
|-|P|-|--|--|--|--|R|R|--|--|A|U|U|--|L|L|--|X|X|--|S|S|
\->ASECU| |  |  |ANY|AC|ANY|AC|ANY|AC|ANY|AC|ANY|AC|ANY|AC|
|-|--|-|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
                        \----------------------------/
                                    |
                              FILE SECURITY
```

P   PURGE flag
S   If 1, system level UDCs exist (only in "SYS" account)
A   If 1, account level UDCs exist for account
E   Account Password Encrypted
U   Account Password Required

Group Entry (Z51 Words)

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

P = PURGE FLAG

Group Entry (Cont.)

```
27|-----------------|23 GROUP CAPABILITY
  | GCAPABILITY     |
30|-----------------|24 GROUP DIR. BASE LINKAGE
  | GLINKAGE        |
31|-----------------|25 GROUP VOL SET DEFN INDX
  | GVSDIPNTR       |
32| GHVSNAME        |26 HOME VOL SET NAME
33|-                |27
34|                 |28 (Definition's acct name)
  | GHVSANAME       |
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|-----------------|38 SAVE CELL FOR GFIPNTR
  | GSAVEFIPNTR     |
47|-----------------|39 GROUP BIND COUNTER
  | GMOUNTREFCNTR   |
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; MVS is in System Domain
            (0:1) = 1; MVS 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 |G |R  |R |R |R |R |A |A |A |A |A |U |U |U |U |
25 |  |  |ANY|AC|AL|GU|GL|ANY|AC|AL|GU|GL|ANY|AC|AL|GU|
   |--|--|---|--|--|--|--|--|--|--|--|--|--|--|--|--|
   |U |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|
   |--|--|---|--|--|--|--|--|--|--|--|--|--|--|--|--|
```

G - Group Password Encrypted

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 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
       |-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|
     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 BLANKS)
    15|           UHGROUP              |13
    16|                                |14
    17|                                |15 /-LOG CNT (N OF USERS
      |--------------------------------|   | LOGGED ON). INIT TO 1
    20|         ULOGCOUNT              |16<  FOR MANAGER.SYS SO THIS
      |-|-|-|-|-|----------------------|   \ USER CANNOT BE PURGED.
UMAXJOBW 21|P|U|E|R|X|W|   JOBPRI      |17 MAX. JOB PRI
      |-|-|-|-|-|----------------------|
    22|       COMM FILE REC 3          |18
      |       (COMMAND FILE LOC OF     |
      |          USER UDC'S)           |
      |--------------------------------|
```
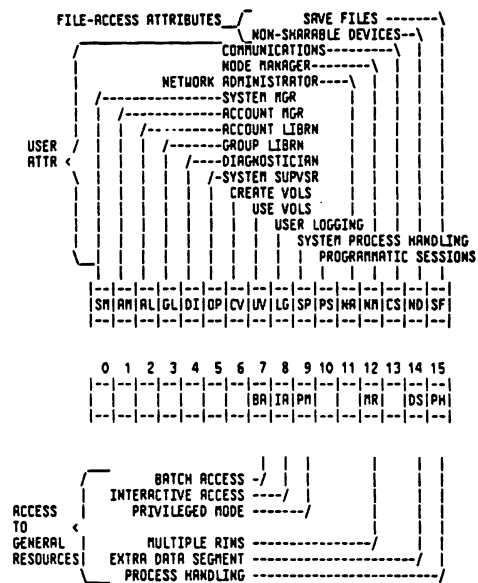
```
P = PURGE FLAG
U = UDC EXIST FLAG
E = USER PASSWORD ENCRYPTED
R = USER PASSWORD REQUIRED
X = USER PASSWORD EXPIRED
W = USER PASSWORD WARNED FOR EXPIRED
```

G.23.00
4- 17

User Attributes/Capabilities

```
FILE-ACCESS ATTRIBUTES_/        SAVE FILES -------\
                      \_ NON-SHARABLE DEVICES--\  |
                  _____ COMMUNICATIONS--------\ | |
                 |         MODE MANAGER--------\| ||
                 |    NETWORK ADMINISTRATOR----\||||
                 |   /--------------SYSTEM MGR-\|||||
         USER /  | /-----------ACCOUNT MGR--\||||||
         ATTR <  | /---------ACCOUNT LIBRN--\|||||||
                 |/---------GROUP LIBRN-----\||||||||
                 |/----------DIAGNOSTICIAN--\|||||||||
                 |/-SYSTEM SUPVSR-----------\||||||||||
                 |     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 -/ |  |        | |
     ACCESS  |    INTERACTIVE ACCESS ----/  |        | |
     TO      |        PRIVILEGED MODE -------/        | |
     GENERAL <                                        | |
     RESOURCES|          MULTIPLE RINS ---------------/ |
             |        EXTRA DATA SEGMENT ----------------/ |
              \        PROCESS HANDLING -----------------/
```

G.23.00
4- 18

Volume Set Definition Entry

```
        0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
       |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
     0|                                              |0
     1|              GVSNAME                          |1  VOLUME
     2|                                              |2  SET
     3|                                              |3  NAME
      |--|--|----------------------------------------|
TY = 0 4|TY|A|          MVTABX                       |4  GVSLINKAGE
      |--|--|----------------------------------------|
     5|VOL COUNT|        VMASK                       |5  GVSINFO
      |------------------------------------------------|
    / 6|                                              |6  MEMBER VOL.
    | 7|                                              |7  NAME(1ST
VOLUME |10|         GVSVOLUME                          |8  ENTRY IS
ENTRY 0 <11|                                          |9  MASTER VOL)
(6 WORDS)|                                            |
    |12|                                              |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.
    63|                                              |51   NAME
      |------------------------------------------------|
    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      VTABX: Volume Table Index
   = 1: Volume Class               VOL COUNT: No. of Volumes
MVTABX: Mounted Volume Table Index (If Mounted)
VMASK: Volume Mask
M = 0: Not mounted
M = 1: Mounted
```

G.23.00
4- 19

G V S L I N K A G E

```
  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
```

G V S I N F O

```
  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 ...
```

G.23.00
4- 20

G V S V O L F L A G S

```
  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

G V S V O L I N F O

```
  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|-  GVCNAME                   |1
 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
10|-  GVCPANAME                 |8   ACCOUNT OF PARENT DEFINITION
11|-                            |9
  |------------------------------|
12|                             |10
13|                             |11
14|   GVCPGNAME                 |12  GROUP OF PARENT DEFINITION
15|                             |13
  |------------------------------|
16|                             |14
17|   GVCPVSNAME                |15  VSNAME OF PARENT DEFINITION
20|-                            |16
21|-                            |17
  |------------------------------|
22|   0                         |18
23|   0                         |19
  |------------------------------|
  ~                             ~
67|   0                         |55
  |------------------------------|
```

G V C L I N K A G E

```
  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

G V C I N F O

```
  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
```
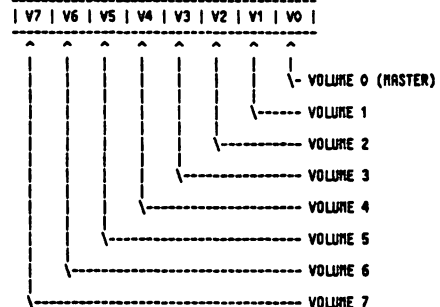
0: NOT MOUNTED OR NON-MEMBER   1: MOUNTED OR MEMBER

CHAPTER 5 LOCK RESOURCES

SIR # Allocation DST X53

SIRs 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
.	.	.
.	.	.

SIRs 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	SPOOL
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 17/18 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. When a process is added to the end of the queue, the priority of the holder of the SIR and the priority of all intervening processes are increased. They are increased to the priority of the newly requesting process.

To get SIRs, arrange the SIRs in ascending order by rank. To release SIRs arrange the SIRs in descending order by rank. For example:

Get SIRs	Release SIRs
GETSIR (LDT) **Rank=85**	RELSIR (ODD) **Rank=92**
GETSIR (ODD) **Rank=92**	RELSIR (LDT) **Rank=85**

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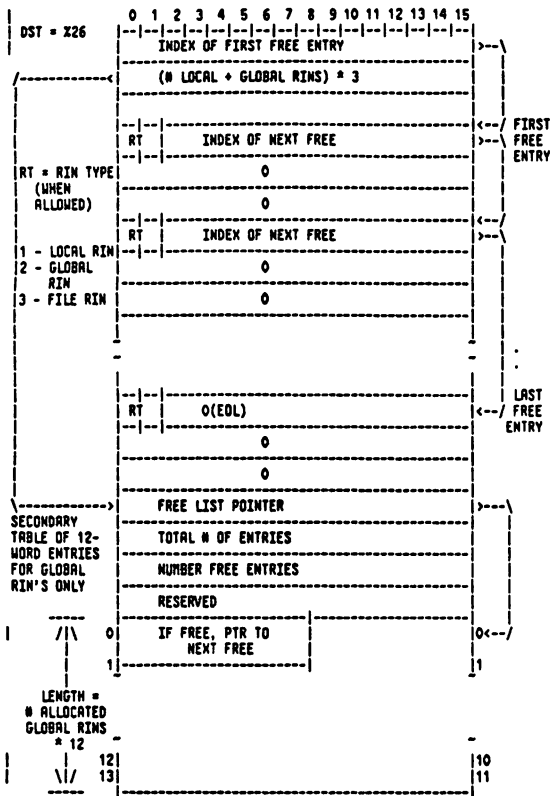
P = PIN #
PIN = PCB table entry number
SIR QUEUE LENGTH - number of processes queued for this SIR

The SIR table is indexed by SIR#, with each SIR# corresponding to a unique, preassigned 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.
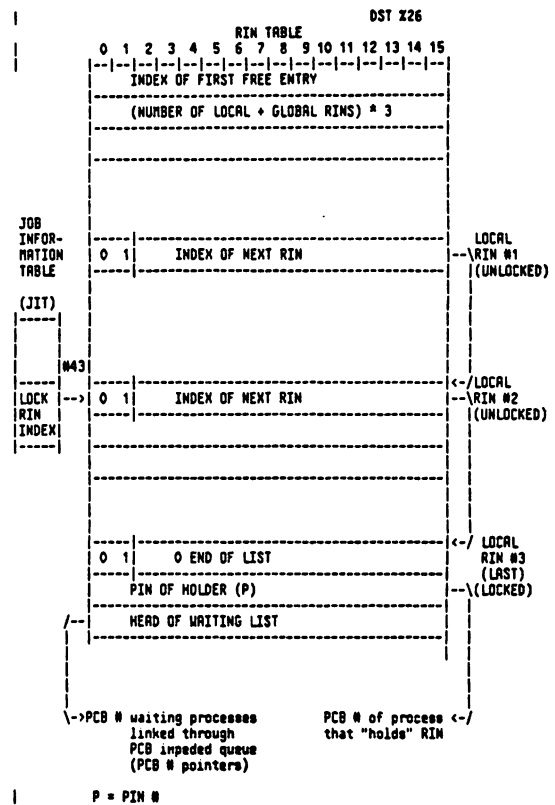
RIN Table General Layout (Initialized State)
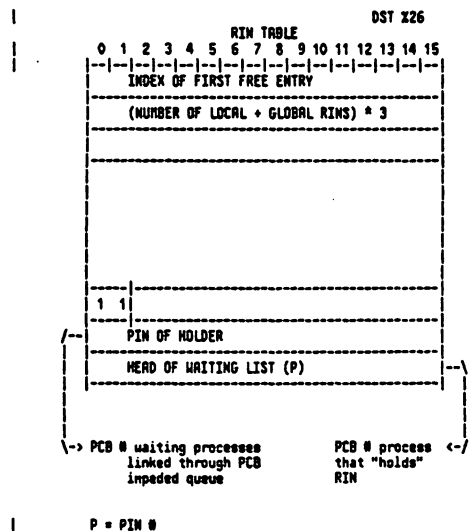
```
                              0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
          |                 |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
          | DST = X26       |        INDEX OF FIRST FREE ENTRY          |>--\
          |                 |-------------------------------------------|
       /--------------------<       (# LOCAL + GLOBAL RINS) * 3         |   |
       |  |                 |-------------------------------------------|   |
       |  |                 |--|--|                                     |<--/  FIRST
       |  |                 |RT|  |      INDEX OF NEXT FREE              |>--\  FREE
       |  |                 |--|--|-------------------------------------|   |  ENTRY
       |  |RT = RIN TYPE    |                    0                      |   |
       |  | (WHEN           |-------------------------------------------|   |
       |  | ALLOWED)        |                    0                      |   |
       |  |                 |--|--|-------------------------------------|<--|
       |  |                 |RT|  |      INDEX OF NEXT FREE              |>--/
       |  |1 - LOCAL RIN    |--|--|-------------------------------------|
       |  |2 - GLOBAL       |                    0                      |
       |  |    RIN          |-------------------------------------------|
       |  |3 - FILE RIN     |                    0                      |
       |  |                 |-------------------------------------------|
       |  |                 -                                           -    :
       |  |                                                                  :
       |  |                 |--|--|                                     |    LAST
       |  |                 |RT|  |      O(EOL)                         |<--/ FREE
       |  |                 |--|--|-------------------------------------|     ENTRY
       |  |                 |                    0                      |
       |  |                 |-------------------------------------------|
       |  |                 |                    0                      |
       |  |                 |-------------------------------------------|
       \--------------------->|        FREE LIST POINTER                |>---\
          SECONDARY         |-------------------------------------------|   |
          TABLE OF 12-      |         TOTAL # OF ENTRIES                |   |
          WORD ENTRIES      |-------------------------------------------|   |
          FOR GLOBAL        |         NUMBER FREE ENTRIES               |   |
          RIN'S ONLY        |-------------------------------------------|   |
                            |               RESERVED                    |   |
      |    /|\    0|        IF FREE, PTR TO         |            |0<--/
      |     |      |        NEXT FREE               |            |1
      |     |     1|--------------------------------------------
          LENGTH =
          # ALLOCATED
          GLOBAL RINS
           * 12      -                                           -
      |     |     12|                                           |10
      |    \|/    13|-------------------------------------------|11
                   ----
```

G.23.00
5- 5

Allocation and Locking of Local RINS

```
                                                         DST X26
       |                            RIN TABLE
       |          0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
       |        |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
                |        INDEX OF FIRST FREE ENTRY          |
                |-------------------------------------------|
                |      (NUMBER OF LOCAL + GLOBAL RINS) * 3   |
                |-------------------------------------------|
                |                                           |
                |-------------------------------------------|
                |                                           |
     JOB        |-------------------------------------------|
     INFOR-     |0 1|        INDEX OF NEXT RIN              |      LOCAL
     MATION     |---|-----------------------------------|   |--\RIN #1
     TABLE      |                                           |   |(UNLOCKED)
                |-------------------------------------------|   |
     (JIT)      |                                           |   |
     |-----|    |-------------------------------------------|   |
     |     |#43 |                                           |   |
     |-----|    |-------------------------------------------|   |
     |LOCK |--->|0 1|        INDEX OF NEXT RIN              |   |<--/LOCAL
     |RIN  |    |---|-----------------------------------|   |--\RIN #2
     |INDEX|    |                                           |   |(UNLOCKED)
     |-----|    |-------------------------------------------|   |
                |                                           |   |
                |-------------------------------------------|   |
                |0 1|        0 END OF LIST                  |   |<--/ LOCAL
                |---|-----------------------------------|   |     RIN #3
                |         PIN OF HOLDER (P)                 |   |--\ (LAST)
                |-------------------------------------------|   |   (LOCKED)
             /--|         HEAD OF WAITING LIST             |   |
             |  |-------------------------------------------|   |
             |
             \->PCB # waiting processes      PCB # of process <-/
                   linked through            that "holds" RIN
                   PCB impeded queue
                   (PCB # pointers)
       |     P = PIN #
```

G.23.00
5- 6

Allocation and Locking of File RINS

```
       |                                      DST X26
       |                     RIN TABLE
       |       0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
       |     |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
             |        INDEX OF FIRST FREE ENTRY          |
             |-------------------------------------------|
             |     (NUMBER OF LOCAL + GLOBAL RINS) * 3    |
             |-------------------------------------------|
             |                                           |
             |-------------------------------------------|
             |                                           |
             |-------------------------------------------|
       |--|  |1 1|                                       |
       |     |---|---------------------------------------|
       |--|  |         PIN OF HOLDER                     |
       |     |-------------------------------------------|
       |     |         HEAD OF WAITING LIST (P)         |--\
       |     |-------------------------------------------|  |
       |                                                    |
       \-> PCB # waiting processes     PCB # process <-/
             linked through PCB        that "holds"
             impeded queue             RIN

       |     P = PIN #
```

G.23.00
5- 7

Allocation and Locking of Global RINS

```
       |                                        DST X26
       |                     RIN TABLE
       |       0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
       |     |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
             |        INDEX OF FIRST FREE ENTRY          |
             |-------------------------------------------|
             |     (NUMBER OF LOCAL + GLOBAL RINS) * 3    |
             |-------------------------------------------|
             |                                           |
             |-------------------------------------------|
             |1 0|     INDEX OF PASSWORD, USERNAME       |----------\
             |---|---------------------------------------|          |
             |         PIN OF HOLDER                     |--\       |
             |-------------------------------------------|  |       |
          /--|         HEAD OF WAITING LIST (P)         |  |       |
          |  |-------------------------------------------|  |       |
          |                                             v          v
          v                                           PCB #      PCB #
        PCB #                                         process
                                                       that
        waiting                                       "holds"
        processes                                      RIN
          -                                             -
             |-------------------------------------------|          |
             |                                           |<---------/
             |-------------------------------------------|
             |         RIN PASSWORD                       |
             |-------------------------------------------|
             |         USERNAME                           |
             |           (USER NAME AND ACCOUNT)          |
             |-------------------------------------------|

       |     P = PIN #

       |     INDEX OF PASSWORD = RELATIVE TO BASE OF SECONDARY
                                 TABLE
```
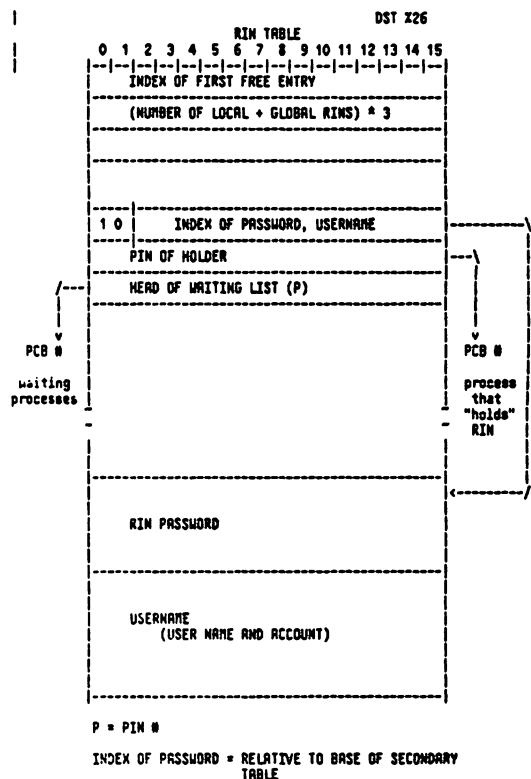
G.23.00
5- 8

CHAPTER 6 FILE SYSTEM

File System Overview

This chapter describes the MPE V file system, including the basic concepts and the table structures used.

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 in this chapter). 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.

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. Therefore, 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 (RFT), 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.

RFT entries can also reside in a global RFT extra data segment. If the file was opened Global RFT (specified in the AOPTIONS) and the program is privileged, then the RFT is placed into this global RFT DST. Any accesses to the file are identical to local RFTs. 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 RFT 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

Below is 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      |(VARIABLE)
|     TABLE           |
|                     |
|---------------------|
|                     |
|   AVAILABLE         |(VARIABLE)
|                     |
|---------------------|
|                     |
|  ACTIVE FILE        |(VARIABLE)
|     TABLE           |
|                     |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  2  3  4  5  6  7  8  9 10 11 12 13 14 15
|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
| 0| PXFILE SIZE IN WORDS                         |0  PXFSIZE
|  |----------------------------------------------|
| 1| LAST DOPEN ERROR NO.  |  LAST COPEN ERROR NO.|1
|  |--|-------------------------------------------|
| 2| N|                                           |2
|  |--|-------------------------------------------|
| 3| LAST DS RFT                                   |3
|  |----------------------------------------------|
| 4| SLAVE RFT NUMBER                              |4
|  |----------------------------------------------|
| 5| LAST KOPEN ERROR NO.  |  LAST FOPEN ERROR NO.|5
|  |-----------------------|----------------------|
| 6| RFT SIZE IN WORDS                             |6  PXRFTSIZE
|  |----------------------------------------------|
| 7|                                               |7
|  | CS TRACE FILE INFO                            |   (PXCTRINFO)
|10|                                               |8
|  |----------------------------------------------|
|11| LAST RESPONDING NO-WAIT I/O RFT ENTRY NUMBER |9  PXFLEFTOFF
|  |----------------------------------------------|
|12| 1ST USER (NOBUF) CONTROL BLOCK TABLE DST NO. |10 PXFCBT1
|  |----------------------------------------------|
|13| 2ND USER (NOBUF) CONTROL BLOCK TABLE DST NO. |11 (PXFCBT2)
|  |----------------------------------------------|
|14| 3RD USER (NOBUF) CONTROL BLOCK TABLE DST NO. |12 (PXFCBT3)
|  |----------------------------------------------|
|15| 4TH USER (NOBUF) CONTROL BLOCK TABLE DST NO. |13 (PXFCBT4)
|  |----------------------------------------------|
|16| 5TH USER (NOBUF) CONTROL BLOCK TABLE DST NO. |14 (PXFCBT5)
|  |----------------------------------------------|
|17| 6TH USER (NOBUF) CONTROL BLOCK TABLE DST NO. |15 (PXFCBT6)
|  |----------------------------------------------|
|20| 7TH USER (NOBUF) CONTROL BLOCK TABLE DST NO. |16 (PXFCBT7)
|  |----------------------------------------------|
|21| 8TH USER (NOBUF) CONTROL BLOCK TABLE DST NO. |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 CBs 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-10
|     ENTRY 1         |(FIXED)
|                     |DL-5
|---------------------|
```
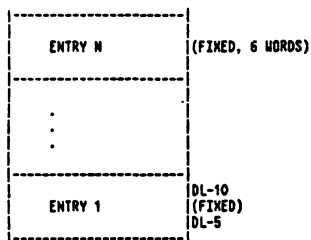
| Where N = PXFAFTSIZE / 6.

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-10 corresponds to file number 1, the entry at DL-16 corresponds to file number 2, etc.

The structure of the global AFT DST is as follows:

```
|---------------------|
|                     |DB + 0
|  ENTRY 0, NOT USED  |
|                     |
|---------------------|
|                     |DB + 6
|     ENTRY 1         |
|                     |
|---------------------|
-                     -
|---------------------|
|                     |DB + (N * 6)
|     ENTRY N         |
|                     |
|---------------------|
```
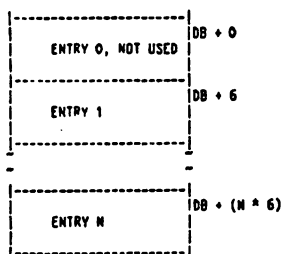
The structure of a file system AFT entry is:

```
| 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
| ENTRY TYPE| N|                         |0
|-----------|--|------------------------ |
|   PHYSICAL ACB DST NUMBER               |1  AFTPACBDST  \
|-----------------------------------------|              > PACB
|   PHYSICAL ACB ENTRY ADDRESS            |2  AFTPACBENTRY_/ VECTOR
|-----------------------------------------|
|   LOGICAL ACB DST NUMBER                |3  AFTLACBDST  \
|-----------------------------------------|              > LACB
|   LOGICAL ACB ENTRY ADDRESS             |4  AFTLACBENTRY_/ VECTOR
|-----------------------------------------|
|   NO-WAIT I/O IOQX                      |5  AFTIOQX
|-----------------------------------------|
```

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.

RFTPACBENTRY   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.

RFTTYPE   This is the RFT 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 (Autodial)
                  6 - KSAM
                  8 - Message File
                  9 - RFA Port
                 13 - Advanced Network Subsystem
```

Remote File RFT Entry

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

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

DS RFT 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 RFT POINTER            |4
|------------------------------------|
|    IOQX                            |5
|------------------------------------|
```

RFT 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 RFT.
   P   - On if PTOP related.
   R   - On if remote main process.
RFT 1   - DS data segment table pointer.
RFT 2   - DSDSCB Index - DS data segment control block index.
RFT 3   - Logical device number.
RFT 4   - Preceding DS open RFT Pointer.
RFT 5   - IOQX - Same as described above.

KSAM RFT Entry

```
 0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
|  FSTYPE  |     UNUSED              |0
|----------|-------------------------|
|    RFT NUMBER OF KEY FILE          |1
|------------------------------------|
|    RFT NUMBER OF DATA FILE         |2
|------------------------------------|
|  KSAM XDS DST (Tagged "KSAM")      |3
|------------------------------------|
|  KSAM XDS DST (If<>0, tagged "RLKSAM") |4
|----------------------------------------|
|    IOQX                            |5
|------------------------------------|
```

RFT 0.(0:4) - FSTYPE (6)
RFT 1   - RFT number of key file
RFT 2   - RFT number of data file
RFT 3   - KSAM XDS DST (Tagged "KSAM")
RFT 4   - KSAM XDS DST (If non-zero, the tag will be "RLKSAM")
RFT 5   - No wait I/O Queue Index

RFT for RFA

```
 0  1  2  3  4  5  6  7  8  9  10  11  12  13  14  15
|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
|0| FTYPE   |   SUBSYS   |   SUBTYPE      |N|F|
|-|---------|------------|----------------|-|-|
|1| RFA'MASTER #         |   LFCB# for RFA    |
|-|---------------------|---------------------|
|2| DSTX FOR RFA XDS BUFFER SPACE           |
|-|-----------------------------------------|
|3| IPC ID, N/W RFA     |  REMOTE ENVNUM      |
|-|---------------------|---------------------|
|4| PENDING FCLOSE DISPOSITION CODE         |
|-|-----------------------------------------|
|5| IOQX (NOWAIT I/O)                       |
|-|-----------------------------------------|
```

RFT 0
   FTYPE   - This field will be 9. Data Comm FTYPE.
   SUBSYS - This field will be 2. RDS application services.
   SUBTYPE - 1 = Remote File Access.
          4 = Remote Data Base Access.
   N   - 0 = File is regular waited.
          1 = File I/O is nowait.
   F   - 0 = An error/failure occurred.
          1 = No error. Normal operation.
RFT 1
   RFA'MASTER # - Buffer number of RFA Master Entry.
   LFCB #   - Buffer number of Local File Control Block Entry.
RFT 2   - DST number of RFA XDS.
RFT 3
   IPC ID   - IPC ID for RFA nowait I/O.
   ENVNUM   - Environment number of remote environment.
RFT 4   - Pending FCLOSE disposition code.
RFT 5   - IOQX - 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
            RFT to CSIOWAIT.

CS Line Entry

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

RFT 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.
RFT 1   - Logical device number of the line.
RFT 2   - Vector to Multiple IOQ indices.
RFT 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.
RFT 4   - DST number of the line's misc data segment.
RFT 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 RFT to CSIOWAIT.

File Control Block Table (CBTAB)

A file control block table can be located in two places: as a subpart of the
PXFILE area, or 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.

There are three types of extra data segment control blocks; expandable,
nonexpandable, and shared FCB. Nonexpandable CBTs are used for a single PRCB
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) CBTs are used for small control blocks ,as LACBs, PACBs with no
buffers, and FCBs which are local to a single process. A list of the
expandable CBTs associated with a process is kept in the overhead area of
PXFILE. When a small control block is needed, these CBTs are checked in
order to see if one of them has room. Shared FCB CBTs are similar to
expandable CBTs except that they belong to the system rather than to a single
process; the system keeps a list of DSTs 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 3 4 5 6 7 8 9 10 11 12 13 14 15
    |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
    |     TABLE SIZE IN WORDS                    |0  CBTSIZE
    |---------------------------------------------|
    |     DST NUMBER CONTAINING TABLE            |1  CBTDSTX
    |-----|---------------------------------------|
    | TYPE|     VECTOR TABLE SIZE IN WORDS       |2
    |--|--|---------------------------------------|
    |  | L|                                      |3  CBTCONTROL
    |--|--|---------------------------------------|
    |     LOCK PCBPTR (PCB'NUM * PCB'SIZE)       |4  CBTPIN
    |---------------------------------------------|
    |     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.(1:1)   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
                PRCB for $STDIN/$STDLIST.

CBTPIN   This is the PCBPTR of the process that has the control block
                locked (PCBPTR = PIN'NUM * PCB'ENTRY'SIZE).

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 PRCB)

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 PINs 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)
|                      |
|----------------------|
```
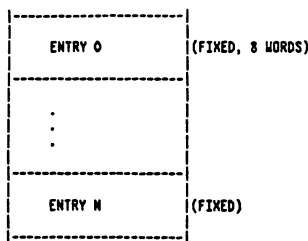
Where N = (CBTVTSIZE/8) - 1.

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 PCBPTR                              |2 VT'PIN  ─    \
|------------------------------------------------|          |
|       HIGH PRIORITY HEAD PCBPTR                |3 VT'QHEAD   |
|------------------------------------------------|          |
|       HIGH PRIORITY TAIL PCBPTR                |4 VT'QTAIL   >-\
|------------------------------------------------|          | |
|       LOW PRIORITY HEAD PCBPTR                 |5 VT'SAVEDHEAD | |
|------------------------------------------------|          | |
|       LOW PRIORITY TAIL PCBPTR                 |6 VT'SAVEDTAIL_/ |
|------------------------------------------------|          |
|       UNUSED                                   |7    PCB   |
|------------------------------------------------|     Relative <--/
                                                      Pointers
```
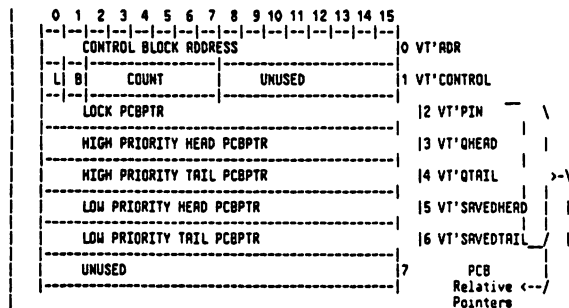
The following partial word identifiers are used:

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

|Discussion: (Note: PIN = PCBPTR in the discussions | PCBPTR
─ PIN'NUM * PCB'ENTRY'SIZE)

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 PRCB 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 PCBPTR of the process which has exclusive access to the control block. Other processes attempting to access the block will be impeded and queued. PCBPTR = (PCB'NUM * PCB'ENTRY'SIZE)

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 VT'QUEUE. The only time this word is used is when the control block is in BREAK mode, which can only happen to an RCB 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 PINs 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  3  4  5  6  7  8  9 10 11 12 13 14 15
|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
| TYPE|      SIZE                                |0 CBDESCRIP
|-----|------------------------------------------|
|                                                |1
|                                                |
|        DATA                                    |
|                                                |
|                                                |
|                                                |N
|------------------------------------------------|
```
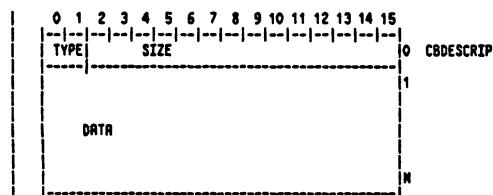
Where N = Size-1.

Partial word field identifiers are:

    CBTYPE   = CB.(0:2)N,   control block type number.
    CBSIZE   = CB.(2:14)N;   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 - PRCB
                    3 - LRCB

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 (RCB)

Virtually every file system intrinsic constructs an RCB as its first action. When using the multi-access option, each accessor shares a single PRCB. 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 RCB.

The PRCB holds information that is global to all accessors of the file. The LRCB holds information that is local to each accessor of the file. At the beginning of a particular access, an RCB is constructed by calling LOC'RCB, which copies information from both the LRCB and the PRCB. At the end of the access, the RCB is released by calling UNLOCK'RCB; this updates the PRCB and LRCB from the RCB 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
  |--|--|--|--|--|--|--|--|--|--|---|---|---|---|---|---|
 0| 3 |      COMPLETE LACB 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
  |-----------------------------------------------------|
10|  RECORD SIZE IN BYTES                               |8
  |-----------------------------------------------------|
11|  BLOCK SIZE IN WORDS                                |9
  |-----------------------------------------------------|
12|  SPARE                                              |10
  |-----------------------------------------------------|
13|  CARRIAGE CONTROL CODE                              |11
  |--|--|--|--|--|--|--|--|--|--|---|---|---|---|---|---|
14| EOF|PG |LN |ST |FK |TC |TB |BB |CAR|DB | EOF T | EOF N  |12
  |--|--|--|--|--|--|--|--|--|--|---|---|---|---|---|---|
15|C |    |   |TE |IC | Q |   |   TERMINAL STOP CHARACTER  |13
  |-----------------------------------------------------|
16|  ERROR CODE                                         |14
  |-----------------------------------------------------|
17|  LAST I/O TRANSMISSION LOG                          |15
  |-----------------------------------------------------|
```

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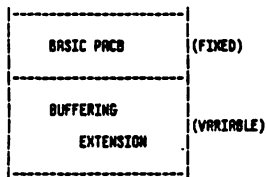
The buffering extension is optional; it is present if and only if the file is accessed with buffering. There are, therefore, two possible formats for an ACB:

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 X63; an ACB "extension" is then generated in words X64 - X67. The resulting ACB thus has the following format:

Access Control Block (ACB) and Physical Access Control Block (PACB)

```
   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
  |-----------------------------------------------------|
10|  RECORD SIZE IN BYTES                               |8
  |-----------------------------------------------------|
11|  BLOCK SIZE IN WORDS                                |9
  |-----------------------------------------------------|
12|  UNUSED                                             |10
  |-----------------------------------------------------|
13|  CARRIAGE CONTROL CODE                              |11
  |--|--|--|--|--|--|--|--|--|--|---|---|---|---|---|---|
14| EOF|PG |LN |ST |FK |TC |TB |BB |CAR|DB | EOF T | EOF N |12
  |--|--|--|--|--|--|--|--|--|--|---|---|---|---|---|---|
15|C |    |   |TE |IC | Q |   |   TERMINAL STOP CHARACTER  |13
  |-----------------------------------------------------|
16|  ERROR CODE                                         |14
  |-----------------------------------------------------|
17|  LAST I/O TRANSMISSION LOG                          |15
  |-----------------------------------------------------|
20|                                                     |16
21|  RECORD TRANSFER COUNT                              |17
  |-----------------------------------------------------|
22|                                                     |18
23|  BLOCK TRANSFER COUNT                               |19
  |-----------------------------------------------------|
24|                                                     |20
25|  FILE POINTER                                       |21
  |-----------------------------------------------------|
26|                                                     |22
27|  CURRENT VARIABLE BLOCK NUMBER                      |23
  |-----------------------------------------------------|
30|                                                     |24
31|  HIGHEST BLOCK NUMBER STARTED                       |25
  |-----------------------------------------------------|
```

Access Control Blocks (ACB's) (Cont.)

	0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15	
32	FCB VECTOR	26
33		27
34	TOTAL NUMBER OF LACB'S	28
35	BX	DEVICE TYPE	LAST LOGICAL I/O STATUS	29
36	LOGICAL DEVICE NUMBER	30
37	PF	HIT		CURRENT BUFFER	TAPE DISPLACE	NO. BUFFERS	31
40	CURRENT RECORD WORD INDEX	32
41	BUFFER SIZE	33
42	VIRTUAL LOGICAL DEVICE NO.	34
43	FMAVT INDEX	35
44	NUMBER OF INPUT LACB'S	36
45	NAME TYPE	FILE DISPOSITION	37
46	ACCESS BIT MAP	BLOCKING FACTOR	38
47	S	M	Q	R	D		AE	RW	ABR	NE	SEOFS	EOFS	39
50	SPOOLED DEVICE TYPE	SPOOLED DEVICE RECORD SIZE	40
51	SPOOLED DEVICE FOPTIONS	41
52	SPOOLED DEVICE AOPTIONS	42
53	IDD OR ODD INDEX	43
54	NO-WAIT DISC ADDRESS	44
55		45
56	UNUSED	46
57	NO-WAIT LOGICAL DEVICE	47
60	P1P2 USED BY FDEVICECONTROL	48
61		49
62	UNUSED	50
63	UNUSED	51

The above words, 0-Z63, are physically located in the PACB of the file. Below, words Z64-Z67, 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, present, will immediately follow word Z63 of the actual ACB in the Control Block Table of the file.

	0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15	
64	DST RELATIVE OFFSET TO PACB	52
65	DST RELATIVE OFFSET TO LACB	53
66	DST RELATIVE OFFSET TO ACB IN THE STACK	54
67	STACK RELATIVE OFFSET TO DB	55

The following identifiers are used when referring to an ACB:

(ACBSIZE)	=	ACB.(2:14)#,	size in words
ACBFNUM	=	ACB(1)#,	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
ACBSTATU	=	ACB(29)#,	access class, status, etc.
ACBBREAK	=	ACBSTATU.(1:1)#,	break ($STDIN/LIST only)
ACBDTYPE	=	ACBSTATU.(2:6)#,	device type
ACBACCCL	=	ACBSTATU.(2:3)#,	device access class
ACBSUBCL	=	ACBSTATU.(5:3)#,	device sub-class
ACBSTATUS	=	ACBSTATU.(8:8)#,	last logical I/O status
ACBQSTATUS	=	ACBSTATU.(8:5)#,	qualifying status part
ACBGSTATUS	=	ACBSTATU.(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 LACBs
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.

ACBDRADDR 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 state word 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
3  - CS80 device
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 pre-spaced. 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 state word 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.

ACBNOWRITEOF 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 ACB. It does not include the buffering extension, if present.

ACBSPROPT   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.

ACBVDRADDR   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 HP 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
    |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
  0|      IOQ ENTRY INDEX      |0 BLXIOQX
    |--------------------------------|--|--|--|--|--|--|
  1|   BLX LDEV NUMBER   |    |U |R |D |W |N |P |1 BLXFLAGW
    |--------------------------------|--|--|--|--|--|--|
  2|        IOCB - STATUS           |2 BLXLSTAT
    |--------------------------------|
  3|     IOCB - TRANSMISSION LOG    |3 BLXTLOG
    |--------------------------------|
  4|                                |4 BLXBLOCK
    |        BLOCK NUMBER            |
  5|                                |5
    |--------------------------------|
  6|                                |6 BLXDRADDR
    |     BLOCK SECTOR ADDRESS       |
  7|                                |7
    |--------------------------------|
 10|                                |8 BLXEXTBASE
    |     BLOCK EXTENT BASE          |
 11|                                |9
    |--------------------------------|
 12|     BLOCK EXTENT SIZE          |10 BLXEXTSIZE
    |--------------------------------|
 13|        UNUSED                  |11
    |--------------------------------|
 14|                                |12 BLXBUFFER
    |                                |
    |        BUFFER                  |
    |                                |
    |--------------------------------|
```

Other identifiers used:

```
    BLXFLAGW      = BLX(1)W,         flag and LDEV word
    BLXLDEV       = BLXFLAGW.(0:8)W, block logical device number
    BLXFLAGS      = BLXFLAGW.(0:8)W, block I/O flags
    BLXUNALLOCEXT = BLXFLAGW.(10:1), block from unalloc. extent
    BLXREVERSE    = BLXFLAGW.(11:1), FREPDBACKWRPD (not used)
    BLXDONTWAIT   = BLXFLAGW.(12:1), I/O status not checked
    BLXIOOUT      = BLXFLAGW.(13:1)W, last I/O was write?
    BLXDIRTY      = BLXFLAGW.(14:1)W, buffer modified?
    BLXIOPEND     = BLXFLAGW.(15:1)W, I/O in progress
    BLXIOCOMP     = BLXFLAGW.(14:2)W, I/O complete - not dirty
    BLXIOCB       = BLXDBL(1)W,       IOCB
```

Discussion:

BLKBLOCK   This is the block number of the data contained in the
                buffer. A value of -1D 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 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 VTABTOLDEV when the label
is read, and converted back by LDEVTOVTAB when the label is written to disc.

The File Control Block has the following format:

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

File Control Block (Cont.)

	0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15	
14	UNUSED	12
15	UNUSED	13
16	END OF DATA POINTER	14 FCBEOF
17		15
20	NO. USER LABELS WRITTEN	NO. USER LABELS AVAIL.	16 FCBUSERLBL
21	EXTENT SIZE IN SECTORS	17 FCBEXTSIZE
22	BLOCKING FACTOR	SECTORS PER BLOCK	18
23	SECTOR OFFSET TO DATA	DISP	NO. EXTENTS-1	19
24	LAST EXTENT SIZE IN SECTORS	20 FCBLAST- EXTSIZE
25	NO. OPENS INPUT MODE	21
26	GROUP NAME - 1ST CHAR.	GROUP NAME - 2ND CHAR.	22 FCBGN
27	GROUP NAME - 3RD CHAR.	GROUP NAME - 4TH CHAR.	23
30	GROUP NAME - 5TH CHAR.	GROUP NAME - 6TH CHAR.	24
31	GROUP NAME - 7TH CHAR.	GROUP NAME - 8TH CHAR.	25
32	ACCT NAME - 1ST CHAR.	ACCT NAME - 2ND CHAR.	26 FCBAN
33	ACCT NAME - 3RD CHAR.	ACCT NAME - 4TH CHAR.	27
34	ACCT NAME - 5TH CHAR.	ACCT NAME - 6TH CHAR.	28
35	ACCT NAME - 7TH CHAR.	ACCT NAME - 8TH CHAR.	29
36	START OF FILE BLOCK NUMBER	30 FCBSTART
37		31
40	CURRENT NUMBER OF DATA BLOCKS IN THE FILE	32 FCBEND
41		33
42	NO. OF OPEN AND CLOSE RECORDS (MESSAGE FILE)	34 FCBNUM- OPENCLSREC
43		35

File Control Block (Cont.)

```
|   |----------------------|----------------------|
|44 | LOGICAL DEVICE NUMBER |                  |36 FCBEXTMAP
|   |----------------------|----------------------|
|45 |    FIRST EXTENT SECTOR NUMBER             |37
|   |----------------------|----------------------|
|   |          .           |                      |
|   |          .           |                      |
|   |          .           |                      |
|   |----------------------|----------------------|
|   | LOGICAL DEVICE NUMBER |                      |
|   |----------------------|----------------------|
|   |     LAST EXTENT SECTOR NUMBER             |
|   |----------------------|----------------------|
```

Other identifiers used:

FCBSIZE	= FCB(0).(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
FCBVERSION	= FCB(4).(9:2)#,	file version (V-Delta-3)
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
FCBMVTRBX	= 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.

FCBEXCLSTART   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

FCBMVTRBX   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 PRCBs 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.

FCBVERSION   Starting with V-Delta-3 this field specifies the version of MPE a file was created on. Legal values:

     0 - a file created before V-Delta-3
     1 - a file created on V-Delta-3 or later
     2,3 - currently undefined

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 if it resides on the first volume, bit 14 if on the second, and so forth.

File Label (FLAB)

The file label has the following format:

```
    0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
   |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
  0| FILE NAME-1ST CHAR. | FILE NAME-2ND CHAR. |0 FLLOCNAME
   |---------------------|---------------------|
  1| FILE NAME-3RD CHAR. | FILE NAME-4TH CHAR. |1
   |---------------------|---------------------|
  2| FILE NAME-5TH CHAR. | FILE NAME-6TH CHAR. |2
   |---------------------|---------------------|
  3| FILE NAME-7TH CHAR. | FILE NAME-8TH CHAR. |3
   |---------------------|---------------------|
  4| GROUP NAME-1ST CHAR.| GROUP NAME-2ND CHAR.|4 FLGRPNAME
   |---------------------|---------------------|
  5| GROUP NAME-3RD CHAR.| GROUP NAME-4TH CHAR.|5
   |---------------------|---------------------|
  6| GROUP NAME-5TH CHAR.| GROUP NAME-6TH CHAR.|6
   |---------------------|---------------------|
  7| GROUP NAME-7TH CHAR.| GROUP NAME-8TH CHAR.|7
   |---------------------|---------------------|
 10| ACCT NAME-1ST CHAR. | ACCT NAME-2ND CHAR. |8 FLACCTNAME
   |---------------------|---------------------|
 11| ACCT NAME-3RD CHAR. | ACCT NAME-4TH CHAR. |9
   |---------------------|---------------------|
 12| ACCT NAME-5TH CHAR. | ACCT NAME-6TH CHAR. |10
   |---------------------|---------------------|
 13| ACCT NAME-7TH CHAR. | ACCT NAME-8TH CHAR. |11
   |---------------------|---------------------|
 14| CREATOR NAME-1ST CHAR. | CREATOR NAME-2ND CHAR. |12 FLUSERID
   |---------------------|---------------------|
 15| CREATOR NAME-3RD CHAR. | CREATOR NAME-4TH CHAR. |13
   |---------------------|---------------------|
 16| CREATOR NAME-5TH CHAR. | CREATOR NAME-6TH CHAR. |14
   |---------------------|---------------------|
 17| CREATOR NAME-7TH CHAR. | CREATOR NAME-8TH CHAR. |15
   |---------------------|---------------------|
 20| LOCKWORD-1ST CHAR.  | LOCKWORD-2ND CHAR.  |16 FLLOCKWORD
   |---------------------|---------------------|
 21| LOCKWORD-3RD CHAR.  | LOCKWORD-4TH CHAR.  |17
   |---------------------|---------------------|
 22| LOCKWORD-5TH CHAR.  | LOCKWORD-6TH CHAR.  |18
   |---------------------|---------------------|
 23| LOCKWORD-7TH CHAR.  | LOCKWORD-8TH CHAR.  |19
   |---------------------|---------------------|
 24|                                           |20 FLSECMX
   |        SECURITY MATRIX                     |
 25|                                           |21
   |---------------------|-------------|--|--|
 26| FILE LANGUAGE ATTRIB.|            |SR| S|22
   |---------------------|-------------|--|--|
```

File Label (Cont.)

```
    0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
   |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
 27|      CREATION DATE                         |23 FLCREATE
   |-------------------------------------------|
 30|      LAST ACCESS DATE                      |24 FLLASTACC
   |-------------------------------------------|
 31|      LAST MODIFICATION DATE                |25 FLLASTMOD
   |-------------------------------------------|
 32|      FILE CODE                             |26 FLFILECODE
   |--|-------------|---------------------------|
 33| C|    MVTABX   |         VMASK             |27 FLPVINFO
   |--|--|--|--|----|---------------|-----------|
 34| S| R| L| X| SUBTYPE | DISC TYPE | R/W |28 FLLOCK
   |--|--|--|--|---------|-----------|-----------|
 35|NO. USER LABELS WRITTEN|NO. USER LABELS AVAIL.|29 FLUSERLBL
   |----------------------|---------------------|
 36|                                           |30 FLFLIM
   |      FILE LIMIT                            |
 37|                                           |31
   |-------------------------------------------|
 40|                                           |32 FLFCBVECT
   |      FCB VECTOR                            |
 41|                                           |33
   |-------------------------------------------|
 42|      CHECKSUM                              |34 FLCHECKSUM
   |-------------------------------------------|
 43|      COLD LOAD ID                          |35 FLCLID
   |-------------------------------------------|
 44|      FOPTIONS                              |36 FLFOPTIONS
   |-------------------------------------------|
 45|      RECORD SIZE IN BYTES                  |37 FLRECSIZE
   |-------------------------------------------|
 46|      BLOCK SIZE IN WORDS                   |38 FLBLKSIZE
   |-------------------------|--|--|-----------|
 47|   SECTOR OFFSET         | V | NO. EXTENTS-1 |39
   |-------------------------|--|--|-----------|
 50|   LAST EXTENT SIZE IN SECTORS              |40 FLLASTEXT-
   |-------------------------------------------|    SIZE
 51|   EXTENT SIZE IN SECTORS                   |41 FLEXTSIZE
   |-------------------------------------------|
 52|                                           |42 FLEOF
   |   END OF DATA POINTER                      |
 53|                                           |43
   |-------------------------|-----------------|
 54|  VOLUME TABLE INDEX     |                 |44 FLEXTMAP
   |-------------------------|-----------------|
 55|     1ST EXTENT SECTOR NUMBER               |45
   |-------------------------------------------|
```

File Label (Cont.)

```
    0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
   |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
   |            .                              |
   |            .                              |
   |-------------------------------------------|
   |   VOLUME TABLE INDEX                       |
   |-------------------------------------------|
   |   LAST EXTENT SECTOR NUMBER                |
   |            .                              |
   |            .                              |
   |-------------------------------------------|
108|                                           |154 FLALLOCTIME
   |   FILE ALLOCATION TIME                     |
109|                                           |155
   |-------------------------------------------|
156|   FILE ALLOCATION DATE                     |110 FLALLOCDATE
   |-------------------------------------------|
157|            UNUSED                          |111
   |-------------------------------------------|
160|                                           |112 FLSTART
   |   START OF FILE BLOCK NUMBER               |
161|                                           |113
   |-------------------------------------------|
162|                                           |114 FLEND
   |   BLOCK NUMBER OF END OF FILE              |
163|                                           |115
   |-------------------------------------------|
164|                                           |116 FLNUMOPENCLSREC
   |   NUMBER OF OPEN AND CLOSE RECORDS         |
165|           (MESSAGE FILE)                   |117
   |-------------------------------------------|
166|                                           |118
   |   LAST FILE MODIFICATION TIME              |   FLMODTIME
167|                                           |119
   |-------------------------|-----------------|
170|  Volume Table Index     | Pext (MODA)     |120 FLPEXT'ADDR
   |-------------------------|-----------------|
171|  Pseudo Extent sector number (LODA)        |121
   |-------------------------|-----------------|
172|File label exten. size|Security extension size|122 FLPEXT'SIZE
   |-------------------------|-----------------|
173|            UNUSED                          |123
   |-------------------------------------------|
174| DEVICE NAME-1ST CHAR. | DEVICE NAME-2ND CHAR. |124 FLDEVNAME
   |---------------------|---------------------|
175| DEVICE NAME-3RD CHAR. | DEVICE NAME-4TH CHAR. |125
   |---------------------|---------------------|
176| DEVICE NAME-5TH CHAR. | DEVICE NAME-6TH CHAR. |126
   |---------------------|---------------------|
177| DEVICE NAME-7TH CHAR. | DEVICE NAME-8TH CHAR. |127
   |---------------------|---------------------|
```

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        = FLFB(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
FLVERSION     = FLAB(39).(8:2)#,   file version(>=V-Delta-3)
FLNUMEXTS     = FLAB(39).(11:5)#,  no. extents less 1
FLLABEL       = FLABDBL(22)#,      label VTAB and sector
FLVTAB        = FLAB(44).(0:8)#,   label VTAB index
FLALLOCTIME   = FLABDBL(54)#,      time allocated on this system
FLALLOCDATE   = FLAB(110),         date allocated on this system
FLSTART       = FLABDBL(56)#,      starting block number
FLEND         = FLABDBL(57)#,      ending block number
FLNUMOPENCLSREC= FLABDBL(58)#,     number of open,close records
FLMODTIME     = FLABDBL(59)#,      last time file was modified
FLPEXT'ADDR   = FLABDBL(60)#,      start address of pseudo extent
FLPEXT'SIZE   = FLAB(122)#,        pseudo extent size
  FLFLAB'EXT'SIZE = FLPEXT'SIZE.(0:8)#,
  FLSEC'EXT'SIZE  = FLPEXT'SIZE.(8:8)#
```

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 Double-word 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 checksum is calculated and compared against the value recorded in the file label. Similarly, each time the file label is written to the disc the checksum 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 CALENDAR intrinsic.

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). Valid for variable and message files only.

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:

Filecode	Mnemonic	Explanation
1024	USL	User Subprogram Library
1025	BASD	Basic Data
1026	BASP	Basic Program
1027	BASFP	Basic Fast Program
1028	RL	Relocatable library
1029	PROG	Program File
1030	NMPRG	Native Mode Program
1031	SL	Segmented Library
1032	NMXL	Native Mode External Library
1033	NMRL	Native Mode Relocatable Library
1035	VFORM	View Form File
1036	VFAST	View Fast Forms File
1037	VREF	View Reformat File
1040	XLSAV	Cross Loader ASCII File (SAVE)
1041	XLBIN	Cross Loader Relocated Binary File
1042	XLDSP	Cross Loader ASCII File (DISPLAY)
1050	EDITQ	Edit Quick File
1051	EDTCQ	Edit KEEPQ File (COBOL)
1052	EDTCT	Edit TEXT File (COBOL)
1054	TDPDT	TDP Diary File
1055	TDPQM	TDP Proof Marked QMARKED
1056	TDPP	TDP Proof Marked non-COBOL File
1057	TDPCP	TDP Proof Marked COBOL File
1058	TDPQ	TDP Workfile
1059	TDPXQ	TDP Workfile (COBOL)
1060	RJEPN	RJE Punch File
1070	QPROC	QUERY Procedure File
1080	KSAMK	KSAM Key File
1083	GRAPH	GRAPH Specification File
1084	SD	User Logging Log File
1090	LOG	Self-describing File
1100	WDOC	HPWORD Document
1101	WDICT	HPWORD Hyphenation dictionary
1102	WCONF	HPWORD Configuration File
1103	W2601	HP 2601 Environment File
1110	PCELL	IDS/3000 Character Cell File
1111	PFORM	IDS/3000 Form File
1112	PENV	IFS/3000 Environment File
1113	PCCMP	
1114	RASTR	Graphics Image in RASTR Format
1130	OPTLF	OPT/3000 Log File
1131	TEPES	TEPE/3000 Script File
1132	TEPEL	TEPE/3000 Log File

1133	SAMPL	APS/3000 Log File
1139	MPEDL	MPEDCP/DRP Log File
1140	TSR	HPToolset Root File
1141	TSD	HPToolset Data File
1145	DRAW	Drawing File for HPDRAW
1146	FIG	Figure File for HFDRAW
1147	FONT	Font File for HFDRAW
1148	COLOR	Color Definition File
1149	D48	
1152	SLATE	Compressed SLATE File
1153	SLATW	Expanded SLATE Workfile
1156	DSTOR	Store File for RAPID/3000 Utility DICTDBU
1157	TCODE	Code File for Transact/3000 Compiler
1158	RCODE	Code File for Report/3000 Compiler
1159	ICODE	Code File for Inform/3000 Compiler
1166	MDIST	HPDESK Distribution list
1167	MTEXT	HPDESK Text
1168	MARPA	ARPA Message File
1169	MARPD	ARPA Distribution List
1170	MCMND	HPDESK Abbreviated Commands File
1171	MFRTM	
1173	MEFT	
1174	MCRPT	
1175	MSERL	
1176	VCSF	
1177	TTYPE	Term Type File
1178	TVFC	Term Vertical Format Control File
1192	NCONF	Network Configuration File
1193	NTRAC	Network Trace File
1194	NLOG	Network Log File
1195	MIDAS	
1211	NDIR	ANODE
1212	INODE	INODE
1213	INVRT	
1214	EXCEP	
1215	TAXON	
1216	QUERF	
1217	DOCDR	
1226	VC	VC File
1227	DIF	DIF File
1228	LANGD	Language Definition File
1229	CHARD	Character Set Definition File
1230	MGCAT	Formatted Application Message Catalog
1235		Reserved
1236	BMAP	MAP File
1242	BDATA	BASIC Date File
1243	BFORM	BASIC Field Order File for VPLUS
1244	BSAVE	BASIC Saved Program File
1245	BCNFG	Config. File for default Option BASIC program
1246	BKEY	
1247	NMBSU	Business Basic/XL Program File
1248	NMBDT	Business Basic/XL Data File
1249	CMBBN	Business Basic/V Binary File

1258	PFSTA	Pathflow STATIC File
1259	PFDYN	Pathflow DYNAMIC File
1270	RFDCA	Revisable Form DCA Document
1271	FFDCA	Final Form DCA Document
1272	DIU	Document Interchange Unit File
1273	PDOC	HPWORD/150 Document
1275	DFI	
1276	SRI	
1401	CUPTX	
1421	MAP	HPMRP/3000 Map Specification File
1422	GAL	
1425	TTX	
1428	RDIC	
1429	RSPEC	
1430	RSPCF	
1431	REXEC	
1432	RJOB	
1433	ROUTI	
1434	ROUTD	
1435	PRINT	
1436	RCONF	
1437	RDICN	HPBRW Dictionary File
1438	REXNM	HPBRW Execution File
1441	PIF	
1461	NMOBJ	
1462	PASLB	
1476	TIFF	Tag Image File Format
1477	RDF	Revisable Document Format
1478	SOF	Serial Object File Format
1479	GPH	Chart File for Charting Gallery Chart
1480	GPD	Data File for Charting Gallery Chart
1483	VCGPM	
1484	FRMAT	Formatter
1485	DUMP	Dump File
1486	NWMDO	New Wave Mail Distribution
1491	X4HDR	X.400 Header
1500	WP1	Other WP1
1501	WP2	Other WP2
1502	LO123	Lotus 123 Spreadsheet
1514	FTCF	Forms tester Cmd Spec
1521	DSKIT	HPDesk Intrinsics Transaction
8000 to 8099		Reserved for RPL

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.

FLMODTIME   Last time the file was modified.

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.

FLPEXT'ADDR   This has the disc address of the start of the pseudo extent (ACD) that has been attached to this file. The high order byte contains the volume table index, the remaining 24 bits contains the sector address.

FLPEXT'SIZE   This word holds the size of the pseudo extent. It is broken up into two halves of one byte each. The high order byte holds the size of the file label extension(not currently implemented), and the low order byte contains the size of the ACD (security extension). Both sizes are in sectors. The pseudo extent is partitioned into two extensions, with the security extension always appearing first, and the file label extension appearing after.

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. Valid for variable and message files only.

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.

FLVERSION   Starting with V-Delta-3, this field specifies the MPE version that a file was created on. Legal values:
    0 - a file created before V-Delta-3
    1 - a file created on or after V-Delta-3
    2,3 - currently undefined

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

File Multi-Access Vector Table (FMAVT) (DST X54)

The FMAVT is used to locate shared PACBs 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 FMAVT 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 X200 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 X4000. 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 4 5 6 7 8 9 10 11 12 13 14 15
|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
| 1| G| D|              UNUSED              |0
|--|--|--|-----------------------------------|
|    JIT DST                                 |1 FM'JITDST
|--------------------------------------------|
|    LOGICAL DEVICE    |                     |2 FM'DADDR
|----------------------|                     |
|    DISC ADDRESS                            |3
|--------------------------------------------|
|                                            |4 FM'PACBV
|    PACB VECTOR                             |
|                                            |5
|--------------------------------------------|
```

FM'DEVICE = FMAVT(0).(2:1)W,   Device bit
FM'GLOBAL = FMAVT(0).(1:1)W,   Global multi-access bit
FM'LDEV   = FM'DADDR(0).(0:8)W, 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+X76,      shared CBT DST no.
MONITOR      = SYSDB+X77,      monitoring flag word
MAXSSECT     = SYSDB+X100,     max # spoolfile sectors
NUMSSECT     = SYSDB+X102,     current # spoolfile sectors
EXTSSECT     = SYSDB+X104,     # sectors/spoolfile extent
SPOOLINDEX   = SYSDB+X132,     class spool index
CSIOWAIT     = SYSDB+X135,     CSIOWAIT PLABEL
CCLOSEPLABL  = SYSDB+X140,     CS CCLOSE PLABEL - FPROCTERM
DSCHKPLABL   = SYSDB+X335,     DSCHECK PLABEL
DSOPENPLABL  = SYSDB+X336,     DSOPEN PLABEL
DSCLOSEPLABL = SYSDB+X337,     DSCLOSE PLABEL
SDSLDEVLABEL = SYSDB+X323,     PLABEL for SDSLDEV
MANWCPLABL   = SYSDB+X340,     MANAGEWRITECONV PLABEL
GLOBALAFTDST = SYSGLBEXT+X121  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 X76 (ABSOLUTE X1076) there exists the shared Control Block Table DST number. This DST holds a list of shared CBTs. Shared CBTs 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 CBTs. Also, all terminal PACBs 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 CBTs.

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 DSTs) starts in the next word after the overhead. The system CBTs are created one at a time as needed. Usually, there are only a few DSTs in the list.

```
|  0|------------------------------------|
|   |  TABLE SIZE IN WORDS (X200)        |0
|  1|------------------------------------|
|   |  DST NUMBER OF THIS TABLE          |1
|  2|------------------------------------|
|   |  0                                 |2
|  3|------------------------------------|
|   |  0                                 |3
|  4|------------------------------------|
|   |  0                                 |4
|  5|------------------------------------|
|   |  0                                 |5
|  6|------------------------------------|
|   |  0                                 |6
|  7|------------------------------------|
|   |  0                                 |7
| 10|------------------------------------|
|   |  1ST. SHARED CBT DST NUMBER        |8
| 11|------------------------------------|
|   |  2ND. SHARED CBT DST NUMBER        |9
| 12|------------------------------------|
|   |  .                                 |10
|  -|  .                                 | -
|   |  .                                 |
|   |------------------------------------|
|177|  118TH. SHARED CBT DST NUMBER      |127
|   |------------------------------------|
```

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

RBS(4)	PCB relative index of current process' PCB entry
%1003	SYSGLOB relative 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	0
1	ENTRY LENGTH (%25)	1
2	# OF UNASSIGNED ENTRIES	2
3	TABLE RELATIVE INDEX TO FIRST UNASSIGNED ENTRY	3
4	TABLE RELATIVE INDEX OF LAST FREE ENTRY	4
5	HIGH WATER MARK	5
6	NUMBER OF PRIMARY CONFIGURED ENTRIES (0)	6
7	HEAD OF IMPEDED QUEUE PCB RELATIVE INDEX	7
10	TAIL OF IMPEDED QUEUE PCB RELATIVE INDEX	8
11	NUMBER OF CURRENTLY IMPEDED PROCESSES	9
12	NUMBER OF MAXIMUM IMPEDED PROCESSES (CURRENT)	10
13	CUMULATIVE NUMBER OF IMPEDED PROCESSES (CURRENT)	11
14	0	12
15	0	13
16	0	14
17	0	15
20	0	16
21	0	17
22	0	18
23	0	19
24	0	20

Unassigned PCB Entry Format

0	0	0
1	TABLE RELATIVE INDEX TO NEXT UNASSIGNED ENTRY	1
.	.	
24	%177777	20

Note: Only word 1 and word 20 are valid for an unassigned PCB entry.

Assigned PCB Entry Format

Assigned PCB Entry Format (Cont.)

```
|------------------------------------------------|
16|    INDEX WITHIN CSTBLOCK TABLE (CSTBLK)    |PBXWORDNUM
|------------------------------------------------|
17|    LOGICAL SEGMENT TRANSFORM TABLE         |MAPDST
  |           (LSTT) DST #                     |
|------------------------------------------------|
20|    ADDRESS (PCB RELATIVE) TO PREVIOUS      |PIMPPIN
  |           IMPEDED PCB                      |
|------------------------------------------------|
21|    ADDRESS (PCB RELATIVE) TO NEXT          |NIMPPIN
  |           IMPEDED PCB                      |
|------------------------------------------------|
22|    BREAKPOINT TABLE RELATIVE ENTRY ADDRESS |BPTLINK
|------------------------------------------------|
23|    ADDR (PCB REL) OF NEXT PROCESS          |NQPTR
  |           IN SCHED QUEUE                   |
|------------------------------------------------|
24|    ADDR (PCB REL) OF PREV PROCESS          |PQPTR
  |           IN SCHED QUEUE                   |
|------------------------------------------------|
```

PCB00.(0:1)   SAR - scheduling attention required
    .(1:1)   Bounds Flag - Privilege mode bounds check
    .(2:1)   CRIT - process is critical or with SIR
    .(3:1)   HSIR - process has a sir
    .(4:1)   PIOVR - pseudo interrupt happened when process had
                SIR or was impeded
    .(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 - don't alter priority
    .(14:1)   STOVA - process aborting due to stack overflow
    .(15:1)   RITBK - Request Information Table Break
                (e.g., message awaiting operator response)

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

PCB02.(0:1)   RDB, 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)   STOVRALL 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 on 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 (pause) 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)   TIM, process waiting for a time out (set up by
                system to prevent a process hang due to a
                possible "lost" event)
    .(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

| PCB10.(0:3)   PSIM, pseudo - interrupt mode
                1: hard kill
                2: soft kill
                3: stop
                4: hibernate
                5: escape (Control-Y)
                6: break
                7: normal
    .(3:1)   WSOFT, OK for soft interrupt to wake process
                even though it is waiting on another event
    .(4:2)   OA (origin of activate)
                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
| .(8:1)   SERVE, if set, this process is a DS SERVER process

| PCB11.(0:1)   LIVE, 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, abort - stack overflow has occurred
    .(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

PCB12.(0:15)   EVENTFLAGS, one for each wait class in PCB04
    .(15:1)   WS, wake up waiting switch set if an awake is
                missing (i.e., the event occurred before the process
                has a chance to wait for it)

PCB13.(0:32)   LASTREFSWAPSEG, segment identifier of last
                referenced swappable code segment

PCB15   (QUEUING INFO)
    .(0:1)   DISPQ - dispatcher's scheduling 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

PCB16.(0:16)   PBX, CSTX block map index of process' program

PCB17.(0:16)   MAPDST, DST entry number of the CST mapping
                table

| PCB20.(0:16) PIMPPIN, PCB relative index of previous impeded PIN

| PCB21.(0:16) NIMPPIN, PCB relative index of next impeded PIN

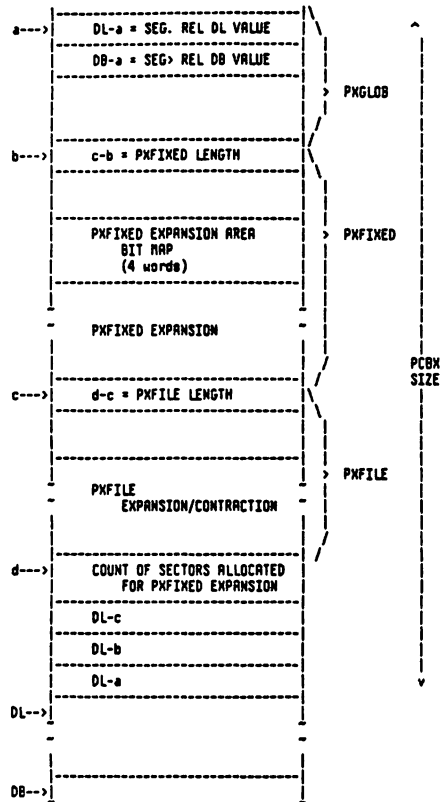| PCB22.(0:16) BPTLINK, breakpoint table relative entry address

| PCB24.(0:16) NQPTR, PCB relative index of next proc in disp queue

| PCB25.(0:16) PQPTR, PCB relative index of prev proc in disp queue
| queue (= -1 if process is not alive)

Process Control Block Extension (PCBX) Structure and Format

Process Control Block Extension (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|  |
        |      BIT MAP          |  |> PXFIXED
        |      (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-->|                       |
```

G.23.00
7- 9

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	EXPAND TO DISPLACEMENT FROM PCBX TO DL	0
1	EXPAND TO DISPLACEMENT FROM PCBX TO DB	1
2	USER ATTRIBUTES (See JIT's UCAP word)	2
3	JMAT INDEX	3
4	JPCNT INDEX	4
5	JCUT INDEX	5
6	SB| R| TY | D| I|GA| | | 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.
GA = Global Allow bit
```

G.23.00
7- 10

PXFIXED Assignments

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

	0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15		
0	C-B PXFIXED SIZE	0	
1	RELATIVE S(S-DB)	1	
2	RELATIVE Z(Z-DB)	2	
3	DB to MORGUE's 0-4	3	
4	INITIAL RELATIVE DL (DB-DL)	4	LM MDST existed LP LOADPROCed
5	GENERAL RESOURCE CAPABILITY (FROM PROG-FILE)	5	Trap Modes .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)-Ctrl-Y
7	LINK TO XDS ENT'S 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| R| ABORT Y |RW| INITIAL CST INDEX	12	< 7:1 = 0 IF HAVE R/W ACCESS
15	MAXIMUM STACK SIZE (MAXDATA LIMIT)	13	TO PROG FILE = 1 OTHERWISE
16	ARITHMETIC TRAP ENABLE MASK	14	8:8 = CST # OF SEG INITIALLY
17	ARITHMETIC TRAP PLABEL	15	EXECUTED AT AT PROC-
20	LIBRARY TRAP PLABEL	16	\ CREATION
21	SYSTEM TRAP PLABEL	17	
22	CONTROL Y PLABEL	18	

NOTE: The General Resource Capability Word (X5) in the PXFIXED
area is used by HP Business Basic. Please inform them
when making any changes to it or to its' location.

G.23.00
7- 11

PXFIXED Assignments (Cont.)

23	CODE TRAP PLABEL	19
24	DATA COMM TERMINATION TRAP PLABEL	20
25	IMAGE TRAP PLABEL	21
26	RESERVED	22
27	CURRENT MAX STACK SIZE (LARGEST VALUE EVER FOR Z-DL)	23
30	PROCESS ELAPSED 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

G.23.00
7- 12

PXFIXED Assignments (Cont.)

```
   | 0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
   |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
52 |  TSLR, VIRTUAL TIME SINCE LAST                |42
   |           RESCHEDULED                         |
   |-----------------------------------------------|
53 |  TSTB, VIRTUAL TIME SINCE TRANSACTION         |43
   |           BEGAN                               |
   |-----------------------------------------------|
54 |  TSSWAPIN, VIRTUAL TIME SINCE SWAPIN          |44
   |-----------------------------------------------|
55 |  TSLA, VIRTUAL TIME SINCE LAST ABSENCE        |45
   |-----------------------------------------------|
56 |  TSLD, VIRTUAL TIME SINCE LAST                |46
   |           DEALLOCATION                        |
   |-----------------------------------------------|
57 |  QCNT, # TIMES TRANSACTION EXCEEDED           |47
   |           THE AST                             |
   |--|--|--|--|--|-----------------------------|
60 |  |D |  |O |  RESERVED FOR FUTURE SOFT        |48
   |  |C |  |S |       INT USE                    |
   |  |Y |  |I |                                  |
   |--|--|--|--|-----------------------------------|
61 |  TRLX INDEX FOR KERNEL TIMEOUT PROCEDURE      |49       JOB TYPE:
   |--|--------------------------------------------|50   1 = SESSION
62 |TY|   JOB/SESSION NUMBER                       |         2 = JOB
   |--|--------------------------------------------|
63 |       RESOURCE COUNT                          |51
   |-----------------------------------------------|
64 | PROCESS ELAPSED CPU TIME (MSEC) SINCE LAST    |52
   |-----------------------------------------------|
65 | CHGROUP COMMAND. USED BY CI'S ONLY.           |53
   |-----------------------------------------------|
66 |     RESERVED FOR FUTURE USE                   |54
   |-----------------------------------------------|
67 |     RESERVED FOR FUTURE USE                   |55
   |--|--|--|--|-----------------------------------|
70 |  |CY|  |SI|                                   |56
   |--|--|--|--|-----------------------------------|
71 |     TIMEOUT TRLX                              |57
   |-----------------------------------------------|
72 |                                               |58
   |-----------------------------------------------|
```

PXFIXED Assignments (Cont.)

```
                                   1  1  1  1  1  1
   | 0  1  2  3  4  5  6  7  8  9  0  1  2  3  4  5
   |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
73 |                                               |59
   |-----------------------------------------------|
74 |     PCLASSMASK                                |60
   |-----------------------------------------------|
75 |     PROCQUESTOPWORD                           |61
   |-----------------------------------------------|
76 |                                               |62
   |     PROCSTOPTIME                              |
77 |                                               |63
   |-----------------------------------------------|
100|     PC CENTRAL TERMINATION DST                |64
   |--------------------|--------------------|--|--|
101|     IFNUM          |   ERROR ON CNT     |V |E |65
   |--------------------|--------------------|--|--|
102|                                               |66
   |- PROC PREEMPT TIME                            |
103|                                               |67
   |-----------------------------------------------|
104|                                               |68
   |     PXFIXED EXPANSION BITMAP                  |
117|                                               |79
   |-----------------------------------------------|
```

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 it got: 1) true bounds violation
                     or 2) an induced bounds violation that
                     indicated that the Control-Y trap procedure
                     may now be entered).

OSI   State of the "USOFT" PCB bit when control-y
               trap was entered. USOFT = 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.

PRIV MODE BOUNDS FLAGS:
   BITS 0-1 = 0 if DB, Q and S bounds checking is disabled
            = 1 if DB bounds checking is disabled with Q and S bounds
                checking enabled
            = 2 if DB bounds checking is enabled with Q and S bounds
                checking disabled
            = 3 if DB, Q and S bounds checking enabled

IFNUM: File number from intrinsic TRACE

ERROR ON CNT: Number time though error on

V = V/Plus transaction trace (1=ON; 0=OFF)

E = Tracing Enable (1=Enable; 0=Disable)

PXFIXED Expansion Bitmap

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

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      |(VARIABLE)
|     TABLE            |
|                      |
|----------------------|
|                      |
|   AVAILABLE          |(VARIABLE)
|                      |
|----------------------|
|                      |
|   ACTIVE FILE        |(VARIABLE)
|     TABLE            |
|                      |DL-5
|----------------------|
```

Overhead

The part labeled Overhead contains mation that pertains to the e section. It
inforis addressed via the pointer at DL-3.
entir

```
                1 1 1 1 1 1
   0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
  |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
 0|     PXFILE SIZE IN WORDS        |0  PXFSIZE
  |--------------------------------|
 1|  LAST DOPEN ERROR # | LAST COPEN ERROR # |1
  |--|                             |
 2|N |                             |2
  |--|--------------------------------|
 3|     LAST DS RFT                 |3
  |--------------------------------|
 4|     SLAVE RFT NUMBER            |4
  |--------------------------------|
 5|  LAST KOPEN ERROR # | LAST FOPEN ERROR # |5
  |--------------------------------|
 6|     RFT SIZE IN WORDS           |6  PXRFTSIZE
  |--------------------------------|
 7|                                 |7
  |    CS TRACE FILE INFO           |   (PXCTRINFO)
10|                                 |8
  |--------------------------------|
11|  LAST RESPONDING NO-WAIT I/O RFT ENTRY # |9  PXFLEFTOFF
  |--------------------------------|
12|  1ST USER (NOBUF) CONTROL BLOCK TABLE DST # |10  PXFCBT1
  |--------------------------------|
13|  2ND USER (NOBUF) CONTROL BLOCK TABLE DST # |11  (PXFCBT2)
  |--------------------------------|
14|  3RD USER (NOBUF) CONTROL BLOCK TABLE DST # |12  (PXFCBT3)
  |--------------------------------|
15|  4TH USER (NOBUF) CONTROL BLOCK TABLE DST # |13  (PXFCBT4)
  |--------------------------------|
16|  5TH USER (NOBUF) CONTROL BLOCK TABLE DST # |14  (PXFCBT5)
  |--------------------------------|
17|  6TH USER (NOBUF) CONTROL BLOCK TABLE DST # |15  (PXFCBT6)
  |--------------------------------|
20|  7TH USER (NOBUF) CONTROL BLOCK TABLE DST # |16  (PXFCBT7)
  |--------------------------------|
21|  8TH USER (NOBUF) CONTROL BLOCK TABLE DST # |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 CBs in PXFILE CBT?
PXFKOPEN	= PXFILE(5).(0:8)#,	last KOPEN error code
PXFFOPEN	= PXFILE(5).(8:8)#,	last FOPEN error code

Discussion:

PXFRFTSIZE	This is the size (in words) of the Active File Table (RFT). 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 RFT 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 RFT is
relocated and the new space is added to the Available Block.

Currently the PXFILE area is only expanded; it is never contracted. For more
information refer to Chapter 6, "File System", and see the Active File Table.

PCBX for Core Resident System Process Stacks

```
    0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
   |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|  \
 0|     DISP FROM PCBX TO DL         |0  \
  |--------------------------------|     |
 1|     DISP FROM PCBX TO DB         |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 PCBN (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,               |DB-9
       |      AND BUSINESS BASIC                   |
       |------------------------------------------|
DB-10| EXTERNAL PLABEL OF OUTER BLOCK             |DB-8
       |------------------------------------------|
DB-7 | RESERVED FOR TRACE AND 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 AND 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
                                   logical unit is initially
                                   referenced.
```

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.

Process Job Cross Reference Table (PJXREF): One DST per system. Used to determine the job/session main process (Command Interpreter) for any process on the system.

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 sharable 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 (JMAT) Structure

```
SIR = 15 = X17
DST = 25 = X31
                                    ENTRY 0
                                      |
                                      v
      0 1 2 3 4 5 6 7 8 910 1 2 3 4 5
     |-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|
   0 |   MAXSIZE    |    CURSIZE     |0  max JMAT size (words/128)
     |--------------|----------------|   current JMAT size (words/128)
   1 |   VMOUNT     |    ENTRY       |1  :VMOUNT state saved for
     |    INFO      |    SIZE        |   WARMSTARTs; JMAT entry
     |              |                |   size (X46)
   2 |     ENTRY POINTER             |2  DB pointer to first entry (X46)
     |------------------------------|
   3 |  WAITING QUEUE HEAD POINTER   |3  DB-relative pointer to entry
     |------------------------------|   at the head of the WAITing list.
   4 |  WAITING QUEUE TAIL POINTER   |4  DB-relative pointer to entry
     |------------------------------|   at the tail of the WAITing list.
   5 |TY|     SCOUNTER              |5  Next assignable session #, TY=1
     |--|---------------------------|   Words 6&7 are a double for
   6 |         NOT USED             |6  MPE X/L compatibility. Only
     |---|--------------------------|   the LSW is used in MPE V/E.
   7 |TY|     JCOUNTER              |7  Next assignable batch #, TY=2
     |------------------------------|
  10 |         NOT USED             |8
     |-|---|-|-|--------|-----------|   L=1, logoff in progress
  11 |L|SEC|S|J| SFENCE | JOBFNCE   |9  S/J=1: S/J # dup check armed
     |-|---|-|-|--------|-----------|   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 DB pointer to head of     Y G
     |------------------------------|   scheduled job queue
```

Job Master Table (JMAT) Structure (Cont.)

```
      |------------------------------|
  17  |       WORKAREA               |15  SFENCE is session fence
  20  |       (23 WORDS)             |16  |
   -  |                              | -  | (ENTRY 0)
  45  |                              |37  v
      |------------------------------|   ---
  46  |                              |38   ^
      |                              |     |
      |                              |     | (ENTRY 1)
      |                              |     |
 113  |                              |75   v
      |------------------------------|    ---
      |                              |     ^
      |                              |     |
      |                              |     | (LAST ENTRY)
   -  |                              | -   |
      |                              |     v
      |------------------------------|    ---
```

SCHEDULING QUEUE

```
WAITING SESSIONS
    FIFO within HIPRI/INPUT priority
[ERROR JOBS        ]
[    FIFO           ]
WAITING JOBS
    FIFO within HIPRI/INPUT priority
```

Job Master Table (JMAT) Entry

```
                   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 |         NOT USED              |2  X70 = scheduled in scheduled
     |------------------------------|       job queue
   3 |                              |3  X40 = waiting, job in
   4 |      USER NAME               |4      scheduling queue
   5 |                              |5  X60 = 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
     |------------------------------|   O = password validated
  17 |                              |15   (STARTDEVICE)
  20 |     GROUP LOGON NAME         |16 1 = must validate password
  21 |                              |17   (INITJSMP)
  22 |                              |18 R = reserved
     |------------------------------|   C = JLIST is device
  23 |       JIN DEVICE             |19   class index
     |------------------------------|
  24 |       JLIST DEVICE           |20
     |------------------------------|
  25 |    JULIAN DATE (CALENDAR)    |21  ty = 1 - session
     |------------------------------|        2 - job
  26 |       TIME (CLOCK)           |22
  27 |                              |23
     |------------------------------|
  30 |    LANGUAGE   |    XPRI      |24
     |---------------|--------------|
  31 |       MAIN PIN              |25
     |------------------------------|
  32 |       CPU LIMIT             |26  0 = default, -1 = no limit
     |-|-|-|------|---------------|
  33 |S|R|N|FT|OUTPRI| NUMCOPIES   |27  ORIGJIN/ORIGJLIST is
     |-|-|-|------|---------------|   used as a scheduling
  34 |       ORIGJIN              |28  link by UCOP when state=
     |------------------------------|   X40 or X70. DB relative ptr.
  35 |       ORIGJLIST            |29  Last entry in list contains
     |------------------------------|   zero (0).
```

Job Master Table (JMAT) Entry (Cont.)

```
      |------------------------------|
  36  |     JMAT CREATOR PIN        |30  Used with the programmatic
      |-|-|-|-----------------------|   creation of sessions/scheduled
  37  |P|W|N|                       |31  jobs.
      |-|-|-|-----------------------|   P = Programmatic logon
  40  |    DATE OF LAST CHGROUP     |32  W = WAITTILLON
      |------------------------------|   N = NOWAIT
  41  |    TIME OF LAST CHGROUP     |33
  42  |                              |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
      R = RESTART                   FT = funny terminal
      N = SEQUENCED                  00 - regular term.
      S = ORIGJIN is spooled.        01 - regular term.,
                                          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	CKSTSTREAM SCHEDULEDSCHED
%40	WAIT	DEVREC JSMP SPOOLER	NURSERY \ SPOOLING /	STARTDEVICE - SCHEDULEJOB SPOOLSTUFFIN ->SCHEDULEJOB
%60	INIT- IALIZAT- ION	UCOP	UCOP	LAUNCHJOB
2	EXEC	JSMP	NURSERY	INITJSMP
3	TERMIN- ATING	JSMP	MORGUE	TERMINATE - EXPIRE - CLEANUPJOB
0	FREE ENTRY	JSMP	MORGUE	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.)

Process Job Cross Reference Table (PJXREF)

DST = %62
TABLESIZE = # PCB entries + 1

```
    0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
   |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
 0|      NUMBER OF ENTRIES                          |
   |-----------------------------------------------|
 1|      J/S NUMBER OF PIN 1                        |
   |-----------------------------------------------|
 2|      J/S NUMBER OF PIN 2                        |
   |-----------------------------------------------|
   ~                                                ~
   |-----------------------------------------------|
 n|      J/S NUMBER OF PIN n                        |
   |-----------------------------------------------|
n + 1|   J/S NUMBER OF PIN n + 1                    |
   |-----------------------------------------------|
```

This table is only used by the SHOWQ command. The entries in the table are set up through PROCREATE and modified by MORGUE.

The job/session number is in the format:

```
    0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
   |------|----------------------------------|   00 = Unused/undefined
   |      |                                  |   01 = Session
   |------|----------------------------------|   10 = Job
                                                 11 = Unused/undefined
                                          Bit 2-15 = Job/Session Number
```

A completely zero entry is either from a system process or a currently unused pin.

Job Process Count Table (JPCNT)
(1 Bit Entry / Running Job)

MEMORY RESIDENT

SYSGLOB BASE = DB+13(%15)
DST = 24 (%30)
SIR = 13 (%15)

```
                         1 1 1 1 1
    0 1 2 3 4 5 6 7 8 9 0 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                        |
   ~    MAXIMUM 64 WORDS LONG       ~
   |                                |
   |-------------------------------|
```

A JPCNT entry must be allocated before the main process can be procreated. The JPCNT Index is located in word 4, PXGLOBAL area, of the stack of a job or session. One JPCNT Index is allocated per job or session.

The job SIR (JIR) = base + JPCNT index, where base is the number of system reserved SIRs. The JIR is used to lock the Job Directory Table.

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 (%44);SIR=14 (%16)
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
 /-|      FREE HEAD                        |2 |        > ENTRIES
 | |-----------------------------------------|        | (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 (%7)*      |2   * May be %10 for System Processes
   |-------------------------------|
 3| POINTER TO ACCT INFO (%60)     |3
   |-------------------------------|
 4| POINTER TO RESERVED (%73)      |4
   |           AREA                |
   |-------------------------------|
 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 $STDIN/X file was
14|   DS DATASEG                  |12            closed w/out encountering an
   |-------------------------------|            EOF. (8:1)=$STDIN,(9:1)=$STDINX
15|   JITASEC                     |13  JITASEC = Account Security
   |-------------------------------|
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
   |       LOGON GROUP            |
   |-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|
   0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
                       1 1 1 1 1 1
```

G.23.00
8- 10

Job Information Table (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   (%65)     |32
   |-|-|---------------------------|
41|P|N| POINTER TO JITGIP  (%67)  |33   P - Group's home volume is
   |-|-|---------------------------|         a private volume
42|   LATTR                       |34   N - Private volume mounted
43|      LOCAL ATTRIBUTES         |35       (i.e., group bound to home
   |-------------------------------|         volume set), JITGIP = %71
44|   PASSF                       |36
45|      PASSED FILE POINTER      |37
   |-------------------------------|
46|   UCAP                        |38
47|      USER CAPABILITY *        |39
   |-------------------------------|
50| RESERVED FOR DS'II            |40
   |-------------------------------|
51|   JITCPUCCHGROUP              |41
52|CPU ms used since last CHGROUP |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
```

G.23.00
8- 11

Job Information Table (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 Information
   |-------------------------------|
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   MVTABX - Mounted Volume
   |-------------------------------|     Table Index
72|   JITGIP                      |58   Group Index Pointer
   |-------------------------------|     Mounted Private Volume Set
73|                            |1 |59
   |-------------------------------|
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
```

* 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|GP|CV|UV|LG|  |PS|NA|NM|CS|ND|SF|
   |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
WORD2|  |  |  |  |  |  |  |BA|IA|PM|  |RR|  |DS|PH|  |
   |--|--|--|--|--|--|--|--|--|--|--|-----|--|--|--|--|
```

G.23.00
8- 12

Allow Mask Format

** The Allow mask for MPE V is expanded to six words. There is a mask in
each user's JIT and the global allow mask in the SYSGLOB extension area.
The Allow mask contains enough bits for a one-to-one correspondence for
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
ALTSPOOLFILE	0	15	15
ALTJOB	1	0	16
BREAKJOB	1	1	17
DELETESPOOLFILE	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

G.23.00
8- 13

	Word	Bit	#
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	9	41
LOG	2	10	42
FOREIGN	2	11	43
IMFCONTROL	2	12	44
SHOWCOM	2	13	45
OPENQ	2	14	46
SHUTQ	2	15	47
DISCRPS	3	0	48

Job Directory Table (JDT)

0	MAX SEGMENT SIZE (WORDS)
1	POINTER TO JDSD
2	POINTER TO JTFD
3	POINTER TO JFEQ
4	POINTER TO JLEQ
5	POINTER TO JJCW
6	POINTER TO FREE SPACE

1 entry per job
DST # in word 10
(base 10) of PXGLOB

```
-                               -
        WORK  AREA
          45 WORDS
-                               -
|--|-----------------------------|
JDSJNUM|TY|      NUM              |
|--------------------------------|
      RESERVED FOR SYSTEM
|--------------------------------|
         JSMPIN
-                               -
JDSD       JOB DATA SEGMENT
              DIRECTORY
-                               -
|                                |
JTFD       JOB TEMPORARY FILE
              DIRECTORY
|                                |
|--------------------------------|
JFEQ       JOB FILE EQUATION
                TABLE
|--------------------------------|
JLEQ       JOB LINE EQUATION
                TABLE
|--------------------------------|
-          JOB CONTROL WORD     -
                TABLE
|--------------------------------|
           FREE SPACE
|--------------------------------|
```

ENTRY SIZE (WDS)	NAME SIZE (WDS)
C1	C2

CN	(X40)

ENTRY INFORMATION

The name is a concatenation of
up to three subnames. Bit 0 of
the 1st character of each
subname is 1.

Job Data Segment Directory Entry (In JDT)

If a DST is allocated as sharable, then it will have entries in both the JDT
and PXFIX. Sharable means that it can be be shared by all processes in the
Command Interpreter process tree (sons, etc.). Nonsharable DSTs only have
entries in the PXFIXED.

```
 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 X2004 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          |     NAME SIZE         |
|       (WORDS)           |      (WORDS)          |
|-----------------------------------------------|
-                                               -
    NAME - ACTUAL FILE DESIGNATOR
|-----------------------|-----------------------|
|     VOLUME POINTER    |                        |
|-----------------------|                        |
|    FILE LABEL POINTER |                        |
|-----------------------------------------------|
```
<-----Name may consist
of up to 4 subnames
(File.Group.Account:
ENVID)

Since all son processes of a CI share the same JDT, exclusive access of the
JDT is controlled with the Job SIR (JIR) and is locked and unlocked by calls
to LOCKJIR and UNLOCKJIR. The JIR number is found in the PXGLOBAL area
(JPCOUNT index). Only job and sessions traces have JIRs, system processes do
not, even though they have JDTs. The JDTs were provided for system processes
for consistency, but are not meant to be increased or reduced.

File Equation Table Entry (In JDT)

```
 0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
|     ENTRY SIZE          |     NAME SIZE         |
|       (WORDS)           |      (WORDS)          |
|-----------------------------------------------|
-                                               -
    NAME - (FORMAL DESIGNATOR)
|                                                |
|     PMASK                                     |*
|                                                |
|-----------------------|-----------------------|
|     NAME LENGTH        |    DEVICE LENGTH       |
|       (BYTES)         |      (BYTES)          |
|-----------------------------------------------|
-    NAME-ACTUAL DESIGNATOR                      -
      (may not be present)
|-----------------------------------------------|
-    DEVICE/CLASS NAME                           -
       (may not be present)
|-----------------------------------------------|
|     FOPTIONS                                  |*
|-----------------------------------------------|
|     AOPTIONS                                  |*
|------------------|------------------|--|--|--|
|     NBUFFERS     |    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         |
|-----------------------------------------------|
|     LANGUAGE (NATIVE LANGUAGE SUPPORT)         |
|-----------------------------------------------|
|  LENGTH     FORMS = / LABEL =                  |
|-----------------------------------------------|
|     FORMS / LABEL     ARRAY                    |
-                                               -
|-----------------------------------------------|
```

Job Line Equation (JLEQ) Entry

```
    |-------------------|-------------------|
    |    ENTRY SIZE     |    DESIG. SIZE    |
    |    (WORDS)        |    (WORDS)        |
    |-------------------|-------------------|
  - |   FORMAL LINE DESIGNATOR              | -
    |      (1-4 WORDS)                      |
    |-------------------|-------------------|
  0 | PMASK1            |                 |0
    |-------------------|---|---------------|
  1 | REF CNT           | 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 |                                     |7
    |                                       |
 10 |                                     |8
    |   DEVICE                              |
 11 |                                     |9
    |                                       |
 12 |                                     |10
    |-------------------|-------------------|
 13 | PMASK3                              |11
    |-------------------|-------------------|
 14 | DRIVER NAME       |                 |12
    |   LENGTH          |                   |
    |-------------------|-------------------|
 15 |                                     |13
    |                                       |
 16 |   DRIVER NAME                       |14
    |                                       |
 17 |                                     |15
    |                                       |
 20 |                                     |16
    |-------------------|-------------------|
 21 | LIST PNTR                           |17
    |-------------------|-------------------|
 22 | COPTIONS                            |18
    |-------------------|-------------------|
 23 | ROPTIONS                            |19
    |-------------------|-------------------|
 24 | DOPTIONS                            |20
    |-------------------|-------------------|
```

Job Line Equation (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 SIZE    |               |
    |   (BYTES)      |               |
    |----------------|---------------|
  - |   NAME                         |
    |----------------|---------------|
    | TY |    MODIFIER               |
    |----|--------------------------|
```

Name may be any alpha-
numeric string, begin-
ning with an alpha,
between 1 and 255 char-
acters long.

TY 00 = OK
    01 = WARN
    10 = FATAL
    11 = SYSTEM

MODIFIER = VALUE FROM 0 TO X377777

Roptions and Foptions Word Breakdown

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

 0 |---|                 0 |---|
   | 0 |                   | 0 |
   |---|                   |---|
   | 0 |                   | 0 |
   |---|                   |---|
   | 0 |                 2 |   | FILE TYPE
 3 |---|                 3 |---|
   |COPY |                 |---|
 4 |---|                   | 0 |
   |NO-WAIT|               |---|
 5 |---|                 5 | 0 | DISALLOW FILES
   |MULTI-               6 |---|
 6 |ACCESS|                | LABELED 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
                   |---|---|
 LABELED TAPE      |   |   | RECSIZE
                   |---|---|
 FRMS MESSAGE      |   |   | DISPOSITION
                   |---|---|
 USER LABELS       |   |   | NUMBUFFERS
                   |---|---|
 LANGUAGE          |   |   | 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)

The UCOP Request Queue (URQ) is used to to signal UCOP that a process is
requesting process deletion. The URQ is a circular queue using a FIFO
algorithm to process requests. When the next available pointer is equal to
the next request pointer, then the table is empty. When the next available
pointer is (logically) one less than the next request pointer and the request
is entered, then the table is full. A full table will cause System Failure 1
(SF1). Thus, the last (logical) entry cannot be used. An entry is added via
a call to REQUCOP.

The UCOP Request Queue (MPE IV) was previously used for many functions such
as stack expansion, but those functions moved to other areas with MPE V. The
only valid entry now is a type 2 entry (process deletion). The original
format is retained in the event that more functions are added.

```
   |----------------------------------------------|
  0|  MAXN 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                                       |
  |  |----------------------------------------------|
  |  |                                              |<-----/
  |  |                                              |
  |  |                                              |
  v  |                                              |
 ---  ----------------------------------------------
```
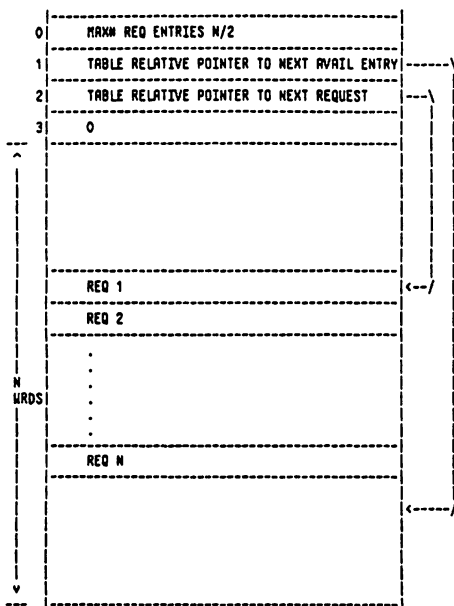
UCOP Entry Format

Each entry is
2 words long

```
I    0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
    |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
    |                                  |      | 2 |
    |--------------------------------------|----------|2 PROCESS DELETION
    |      PIN                                         |
    |--------------------------------------------------|
```
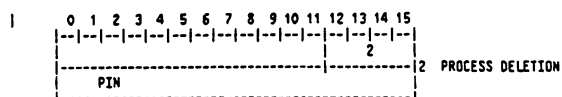
CHAPTER 9 RELOCATABLE OBJECT CODE

USL Files Introduction

- USL record length is always 128 words
- Layout of double-word disc addresses:

```
|----------------------------|-----------------|
|                            |   WORD #        |
|     25-BIT RECORD #        |   WITHIN RECORD |
|----------------------------|-----------------|
0                          24 25             31
```
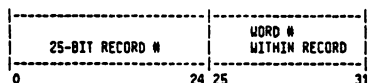
- Hash links join all entries with the same hash key regardless of
  type
- Linear lists terminate with a zero link
- Circular lists containing only the list head point directly to
  themselves
- Single-word disc addresses:

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

Overall USL File Format (Cont.)

```
   |--------------|
 12|   SRAD       |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 0       |33  HASH LINKS
   |--------------|
   |   .          |
   |   .          |
177|   HL 94      |127
   |--------------|
```

USL Files General Information

```
       0|----------|0        ^
        |          |         |
        |  RECORD 0|         |
        |          |         |
     177|----------|127      |
                             |
        /          /         |
        |          |         |
     200|----------|128      |
        |          |         |
  DL <  | DIRECTORY|      32K |
        |  ENTRIES |      MAX |
        \          |          |
        /          |----------|
        |          |    |     |
     SRAD  AVAILABLE|         |
  ADL <   |DIRECTORY|         |
        |          |         |
        \          |          v
        /     SAI*|----------|
        |          |          |
        |          |  INFO    |
     IL |          |(HEADERS) |
        |          |  (CODE)  |
        \          |----------|
        /          |    |     |
        |     SARI |----------|
     AIL |          |AVAILABLE |
        |          |  INFO    |
        \     FL-1 |----------|
```
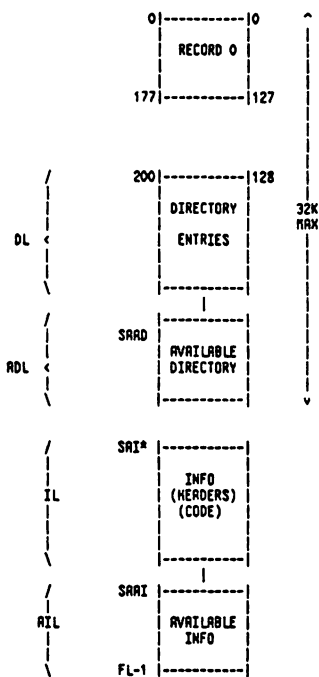
* SAI must be on a record boundary

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     |-\ /|          |
        |----------|  \ |----------| \/ |----------|
                       v      ^       ^
       /----------/---------/----------------------\
       |                                            |
       |  |----------|    |----------|    |----------|
       \->| 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        |
          |----------|    |----------|    |----------|
```
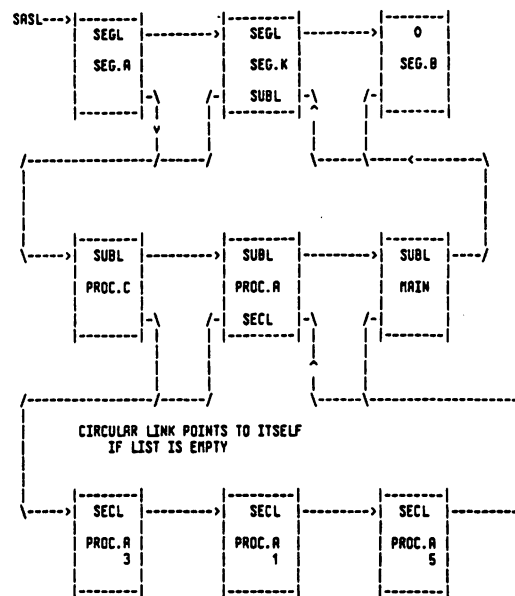
```
A \                        PROC C \
K  > Segment name entries   PROC A >Subprogram
B /                         MAIN  / entries

   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 2 3 4 5 6 7 8 9 10 11 12 13 14 15
| |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|   Number of words in this
| |           NW          |      0          |   block
| |--|-------------------------------------|
| |                                         |
| |              GARBAGE                    |
| |                                         |
| |-----------------------------------------|
```

Entry Type 1

SEGMENT NAME

```
                       1 1 1 1 1 1
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
   |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
  0|         NW          |         1          |
   |--|-----------------------------------------|
  1|              HL                            |
   |--|--|-----------------|-------------------|
  2|A |    NC    |  CHAR1                       |
  3|  :                                         |
   |  (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
1,5	1	1,2,3,4	1,2,3,4
TSDB	TSDB	TSDB	TSDB
NWPUST	NWPUST	NWPUST	NWPUST
5			
NWSDB	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

Entry Type 2 (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
    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
       having the same segment name

SECL - Secondary entry point list link

SSA - Program unit starting PB address

SRC - 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)

NWSDB - Number of words in secondary
        DB array (SPL)

T - Terminating bit - set if last set of
    headers in entry

NH - Number of headers

SRH - 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  9 10 11 12 13 14 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  4  5  6  7  8  9 10 11 12 13 14 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.)

```
|  0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
|     SRC                                       |
|--|--|-----------------------------------------|
|F |W |       NWC                               |
|--|--|-----------------------------------------|
|     SE                                        |
|-----------------------------------------------|
|     TPDB                                      |
|-----------------------------------------------|
|     TSDB                                      |
|-----------------------------------------------|
|     NWPUST                                    |
|-----------------------------------------------|
|     NWD/NWO                                   |
|--|--|----------------------|------------------|
|P |  |        NP            |       CN         |
|--|--|----------------------|------------------|
|     TN                                        |
|-----------------------------------------------|
|     PARM.1                                    |
|-----------------------------------------------|
|                                               |
|        (VARIABLE # OF PARMS. SEE CN)          |
|                                               |
|-----------------------------------------------|
|     PARM. NP                                  |
|--|--------------------------------------------|
|T |   NH                                       |
|--|--------------------------------------------|
|     SRH                                       |
|-----------------------------------------------|
|     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 node bit. Set if procedure is to be executed in privilege node

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

SRC - 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)

Entry Type 4 (Cont.)

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 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 PARM 's and type of each PARM.

NP - Number of PARMs

CN - Character count of PARMs

TN - Procedure Type (see Data Descriptors earlier in this chapter).

T - Terminating bit. Set if last set of headers in entry.

NH - Number of headers

SRH - Starting address of header

HDW - Header (pointer)

Entry Type 5

PROCEDURE - SECONDARY ENTRY POINT

```
  0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 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  4  5  6  7  8  9 10 11 12 13 14 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                                        |
|--------------------------------------------------|
|       SRC                                        |
|--|--|--------------------------------------------|
|F |W |    NWC                                      |
|--|--|--------------------------------------------|
|T |    NH                                          |
|--|------------------------------------------------|
|       SRH                                        |
|--------------------------------------------------|
|       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

SRC - 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

SRH - Starting address of header

HDW - Header (pointer)

Entry Type 7

BLOCK DATA

```
  0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 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                                          |
|--|------------------------------------------------|
|    SRH                                           |
|--------------------------------------------------|
|    HDW                                           |
|--------------------------------------------------|
|       .                                          |
|       .                                          |
|--------------------------------------------------|
|    HDW                                           |
|--------------------------------------------------|
|       .                                          |
|       .                                          |
|       .                                          |
|--------------------------------------------------|
```

Entry Type 7 (Cont.)

```
 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
|    CAL                                         |
|------------------------------------------------|
|       | NC        | CHAR 1                      |
|------------------------------------------------|
|                                                |
|    COMMON ARRAY NAME                            |
|    .                                           |
|------------------------------------------------|
|    CHAR.NC         |                            |
|--|-----------------|----------------------------|
|T |    NH           |                            |
|--|---------------------------------------------|
|    SRH                                          |
|                                                |
|------------------------------------------------|
|    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

SRH - Starting address of headers

HDW - Header (pointer)

Entry Type 8

PROCEDURE - SECONDARY ENTRY POINT

```
 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
|  |    NW            |              8            |
|--|---------------------------------------------|
|    HL                                           |
|--|--|--|-----------|----------------------------|
|A |C |  |H | NC      | CHAR 1                      |
|--|--|--|-----------|----------------------------|
|                                                 |
|    (VARIABLE N 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

CN - Character count of PARMS

TN - Procedure type

Entry Header Format

```
SRH ------>|------------------|
           |     HEADER       |
           |------------------|
           |     .            |
           |     .            |
           |     .            |
           |------------------|
           |     HEADER       |
           |------------------|
SRH ------>|     HEADER       |
           |------------------|
           |     .            |
           |     .            |
           |     .            |
           |------------------|
SRC ------>|     CODE         |
           |------------------|
           |     .            |
           |     .            |
           |     .            |
           |------------------|
           |     HEADER       |
           |------------------|


SRH ------>|------------------|
           |     HEADER       |
           |------------------|
           |     .            |
           |     .            |
           |------------------|
           |     HEADER       |
           |------------------|
```

Each entry (except secondary entry point entries) must describe N > 0 sets of
headers. The headers in each set must be continuous and in the same order as
the HDW 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, SRC = SRH.

If the entry has no header set, then NH, SRH sequence is absent.

Header Type 0

GARBAGE

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

Header Type 1

PCALs

```
  0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 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
      the 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 2 3 4 5 6 7 8 9 10 11 12 13 14 15
|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
| |            NW            |      2      |
|--|------------------------------------|
| |   PBA                                |
|----------------------------------------|
|       .                                |
|       .                                |
|----------------------------------------|
| |   PBA                                |
|----------------------------------------|
```

PBA - PB' address of PB address
      to be corrected

Header Type 3

OWN / DATA VARIABLES

```
  0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
| |            NW            |      3      |
|--|------------------------------------|
| B|   PBA                               |
|--|------------------------------------|
|       .                                |
|       .                                |
|--|------------------------------------|
| B|   PBA                               |
|--|------------------------------------|
```

PBA - PB' address of own variable
      pointer to be corrected

Header Type 4

DSDB / OWN / DATA / VALUES

```
  0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 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 2 3 4 5 6 7 8 9 10 11 12 13 14 15
|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
| |            NW            |      5      |
|--|------------------------------------|
| B|   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 2 3 4 5 6 7 8 9 10 11 12 13 14 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 9 0 1 2 3 4 5
|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|
| |            NW         |    7    |
|-|--------------------------------|
| |   TN                           |
|-|-----------|--------------------|
|M|   NC      |   CHAR.1            |
|-|-----------|--------------------|
|       .                          |
|       .                          |
|----------------------------------|
|   CHAR.NC      |                 |
|----------------------------------|
|   DA                             |
|----------------------------------|
|   PBA                            |
|----------------------------------|
|       .                          |
|       .                          |
|----------------------------------|
|   PBA                            |
|----------------------------------|
```

PBA - PB' address of linked lists
      of instructions to be repair-
      ed; 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 mon-
    itored by DEBUG.

DA - Logical word disp. in PUST;
     lower 8 bits of word will be
     init. with prin.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 0 1 2 3 4 5
|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|
| |         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 9 0 1 2 3 4 5
|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|
| |         NW          |    9    |
|-|---------------------|---------|
|    NWC              |
|--------------------|
|      | NC | CHAR.1  |
|--------------------|
         .
         .
         .
|--------------------|
|   CHAR.NC  |
|-|-|--------|
|B|M|    NL              |
|-|-|--------------------|
|   LD                   |
|------------------------|
|   DA                   |
|------------------------|
|   PBR                  |  ---
|------------------------|   ^
         .                   |
         .                   | NL
         .                   |
|------------------------|   v
|   PBR                  |  ---
|------------------------|
         .
         .
         .
|-|--------------------|
|B|M|    NL            |
|-|-|------------------|
|   LD                 |
|----------------------|
|   DA                 |
|----------------------|
|   PBR                |
|----------------------|
         .
         .
         .
|   PBR                |
|----------------------|
```

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
     prin. 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

PBR - 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
      PBR = -1 indicates NO FIX-UPS

Header Type 10

LOGICAL UNITS

```
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|
| |        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 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|
| |         NW          |   11    |
|-|---------------------|---------|
|   PBR                 |
|-----------------------|
|   NC                  |
|-----------|-----------|
|  CHAR.1   |  CHAR.2    |
|-----------|-----------|
       .
       .
|-----------|-----------|
|  CHAR.NC  |           |
|-----------|-----------|
```

PBR - 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.

Header Type 12

```
|  0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
|  |-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|
| O|X|     NW        |      12     |
|  |-|---------------|-------------|
|  |    ID NUMBER    |
|  |-----------------------------|
```

Sub-Header For Header Type 12

```
|  0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
|  |-|---------------|-------------|
| X|     NW          |   SUBHDR    |
|  |-|---------------|-------------|
|  |     OFFSET      |
|  |-----------------------------|
|  |    FATHER - LINK |              IF SUBRTN. PARM OR LBL, ELSE 0
|  |-----------------|-----------|-|
|  | CLASS   | TYPE  |I|            I = 1 IF INDIRECT
|  |---------|-------|-|            TYPE = 0 IF NO SUBTYPE
|  | MODE    | ADDRESS |
|  |---------|---------|
|  |   BROTHER - LINK  |            OFFSET OF PRECEDING SUB., OR 0
|  |-------|-----------|
|  | NC    |   CHAR 1  |
|  |-------|-----------|
|  |         .          |
|  |         .          |
|  |-----------------|-----|
|  | CHAR NC         |     |
|  |-----------------|-----|-|
|  |                       |SPL|
|  |-----------------------|-|-|
|  |     VALUE       |              VALUE OF EQUATE,
|  |-----------------|              ENDING CODE OFFSET OR PARENT
                                    SUBROUTINE IF SUBRTN PARAMETER,
                                    PB-REL OFFSET IF LABEL,
                                    OTHERWISE 0.
```

Header Type 13 & 14

```
|  0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
|  |-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|
| X|     NW         |13 OR 14 |
|  |-|--------------|-----------|
|  |   ID NUMBER    |
|  |----------------------------|
|  |  FATHER - LINK ( = 0)      |
|  |----------------------------|
|  |  BROTHER - LINK ( = 0)     |
|  |----------------------------|
|  |  SON - LINK ( = 0)         |  OFFSET OF 1ST SUBROUTINE, OR 0
|  |----------------------------|
|  |  NEXT - LINK ( = 0)        |
|  |----------------------------|
|  | NC      |   CHAR 1         |
|  |---------|------------------|
|  |         .                  |
|  |         .                  |
|  |---------|------------------|
|  | CHAR NC |                  |
|  |---------|------------------|
|  |  LEVEL  |   SPL            |  LEVEL = 0 IF TYPE 13, 1 IF 14
|  |---------|------------------|
```

For Header Type 12, 13, & 14:

NW	-	The total length in words of the header.
SUBHDR	-	0 if non-variable, 1 if var., 2 if PB array.
ID NUMBER	-	A unique reference for each block. (A block is a procedure, subroutine, or the outer block.) This number is kept in the 24th word of record zero of the USL file. It is incremented every time a block is processed, and survives from compile to compile as long as the USL file is not initialized.
V	-	Set to 1 if this is a modifiable identifier (i.e., a variable).
OFFSET	-	A number assigned to each identifier within a block. When combined with the ID Number, this uniquely identifies each identifier in a program.

FATHER-LINK	-	The offset of the parent procedure of this subroutine. Since more than one subroutine in a procedure can have the same parameter and label names, the FATHER-LINK serves the same purpose on a local level that the ID Number does globally. When combined, the FATHER-LINK.ID NUMBER.OFFSET combination provides definitive identifier recognition.
BROTHER-LINK	-	The offset of the subroutine preceding this one, (if any).
SON-LINK	-	The offset of the first subroutine (if any) in the procedure or outer block.
CLASS/TYPE	-	The identifier class and subclass (See SPL IMS).
I	-	Set to 1 if a reference parameter.
MODE/ADDRESS	-	The identifier addressing mode & offset (See IMS).
NC	-	The number of characters in the identifier name.
CHAR1..CHARNC	-	The identifier name.
SPL	-	A value (currently 99) identifying the source language to Symbolic Debug.
VALUE	-	See Diagram.
LEVEL	-	See Diagram.

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|       |
|  | SRXL  |  S.A. EXTERNAL SET LIST
| 5|       |5
|----------|
| 6|       |6                        1|--------|
| 7|       |7                          | FREE MAP|
|10|       |8                          |   0    |
|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 word 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:

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

CN- Character count

TN- Procedure Type (see Data Descriptors earlier in this chapter).

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 X17777777777D terminates the list.

Headers

```
      0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
   |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
   |  |           NW              |        1         |
   |--|--|--------------------------------------------|
   |F |W |         NW CODE                            |
   |--|--|--------------------------------------------|
   |                                                  |
   |          S.A. INFO BLOCK                         |
   |                                                  |
   |--------------------------------------------------|
   |          S.A. ENTRY                              |
   |--------------------------------------------------|
   |          PBA                                     |
   |--|--|--|--------------------------------------------|
   |S |U |I |       NC      |       CHAR 1            |
   |--|--|--|--------------------------------------------|
   |                     .                            |
   |                     .                            |
   |--------------------------------------------------|
   |          CHAR.NC       |                         |
   |--------------------------------------------------|
   | LC |      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
    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
CN- Character count
TN- Procedure Type (see Data Descriptors earlier in this chapter).

All headers are the same as in a USL except for the PCAL header.

CHAPTER 10 PREPARED OBJECT CODE

Program File Format

Addr		Word	Description
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	4	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	SRE	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	SAFLWT	12	DB ADR. OF FLWT (-1 IF NO FLWT)
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	
25		21	> Three words reserved for MPE/XL.
26		22	

Program File Format (Cont.)

```
        -              -
     |------------|
  27 |            |23   NOTE: ALL UNUSED WORDS ARE RESERVED FOR
     |------------|           FUTURE USE AND SHOULD BE SET TO
  30 |            |24          ZERO.
     |------------|
  31 |            |25
     |------------|
  32 |            |26
     |------------|
  33 |            |27
     |------|------|
     | CST  | CST  |
  34 |  0   |  1   |28 \
     |------|------|    |
     |            |     |
     | CST        |     > CST REMAPPING ARRAY
     |  n         |     |
     |--|--|------|    /
     |P |S |  SL  |  \
     |  |  |   0  | 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)/2
L = ((28 + NS + (NS + 1)/2 + 127)/128) 128

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

/ [BA] (7) BATCH ACCESS
| [IA] (8) INTERACTIVE ACCESS
| [PM] (9) PRIVILEGED MODE
ACCESS TO
GENERAL   <
RESOURCES
| [MR] (12) MULTIPLE RINS
| [DS] (14) EXTRA DATA SEGMENT
\ [PH] (15) PROCESS HANDLING

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 privileged mode, indexed by segment number. All segments begin on a record boundary. The number of records for a given segment is (SL + 127)/128. The record number, SRS, of segment N (0 <=N <=NS-1) is:

```
    If N > 0 then
    FOR I=0 TO N-1
      BEGIN
      SRS:=SRS + (SL(I) + 127)/128
      END
    ELSE
        SRS:=SRS;
```

Global Area Format

A set of records containing the initial set begins at record SRG (Word 3) and consists of (GS + 127)/128 records.

External List

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
Parm1...Parm NP = Contain Data Descriptors documented
                      in Chapter 9, "Relocatable Object Code"

Entry Point List

NOTE: The entry point list must immediately follow the external
      list.

Code Segment With Patch Area

Patch Area

PMAP Information

```
|----------------|
|                |
|     PTT        |     PMAP TYPE TABLE
|                |
|----------------|
|                |
|     SPP        |     SEGMENT PMAP POINTERS
|                |
|----------------|
|                |
|     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

```
|                     1 1 1 1 1 1
|    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            |
    |------------------------------|
    |           SEGNUM             |
    |------------------------------|
```

Type 1 Procedure PMAP Record

```
|                     1 1 1 1 1 1
|    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

```
|                     1 1 1 1 1 1
|    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

```
|                     1 1 1 1 1 1
|    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
|   |-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|
|  0|         LID            |0        NOTE:  LID = 4 - EXPANDED SYSTEM SL
|   |--------------------------|              = 3 - ACCOUNT OR GROUP SL
|  1|         FL             |1        FILE LENGTH (IN RECORDS)
|   |--------------------------|
|  2|         EL             |2        EXTENT LENGTH (IN RECORDS)
|   |--------------------------|
|  3|                        |3
|   |--------------------------|
|  4|         NSEG           |4        # OF SEGMENTS
|   |--------------------------|
|  5|                        |5
|   |--------------------------|
|  6|                        |6
|   |--------------------------|
|  7|         FRTL           |7        S.A. OF FREE R.T. ENTRY LIST
|   |                        |         (-1 IF NONE)
| 10|                        |8
|   |--------------------------|
| 11|         NRT            |9        # OF REFERENCE TABLE ENTRIES
|   |                        |
| 12|                        |10
|   |--------------------------|
| 13|         NS             |11       # OF SECTIONS
|   |                        |
| 14|         NOT USED       |12
|   |--------------------------|
|15 |         NOT USED       |13
|   |                        |        \
|16 |  YEAR    | DAY OF YEAR |14      |
|   |----------|-------------|        | TIME AND
|17 | HOUR OF DAY | MINUTE OF HOUR|15  > DATE STAMP
|   |----------|-------------|        | OF LAST
|20 |  SECONDS  | TENTH OF   |16      | MODIFICATION
|   |           | SECONDS    |        /
|   |--------------------------|
|   -                        -
|   |--------------------------|
| 41|       HASH LIST        |33
|   |                        |
|   |          .             |
|   -                        -        NOTE: Uninitialized fields are
|   |--------------------------|            reserved for future use and
|177|        HL94            |127          should be zero.
|   |--------------------------|            HL = Hash List.
```

SL File Format (Cont.)

For Group and Account SL   For System SL

```
 0|-------------|          0|-------------|
  |   RECORD    |           |   RECORD    |
  |     0       |           |     0       |
  |-------------|           |-------------|

 1|-------------|   <--REFERENCE    1|-------------|
  |   RECORD    |     TABLE------->  |   RECORD    |
  |     1       |     POINTERS       |     1       |
  |-------------|                    |-------------|
                      \-->2|-------------|
 2|-------------|           |   RECORD    |
  |  FREE MAP   |           |     2       |
  |     0       |           |-------------|
  |-------------|
                          3|-------------|
NS+1|-----------|          |  FREE MAP   |
  |  FREE MAP   |          |     0       |
  |    NS-1     |          |-------------|
  |-------------|
                        NS+2|-----------|
NS+2|-----------|          |  FREE MAP NS-1 |
  |  AVAILABLE  |          |-------------|
  |             |
  |             |       NS+3|-----------|
LAST|-----------|          |  AVAILABLE  |
REC |             |       |-------------|
  | AVAILABLE   |
  |-------------|       LAST|-----------|
                       REC | AVAILABLE   |
                          |-------------|
```
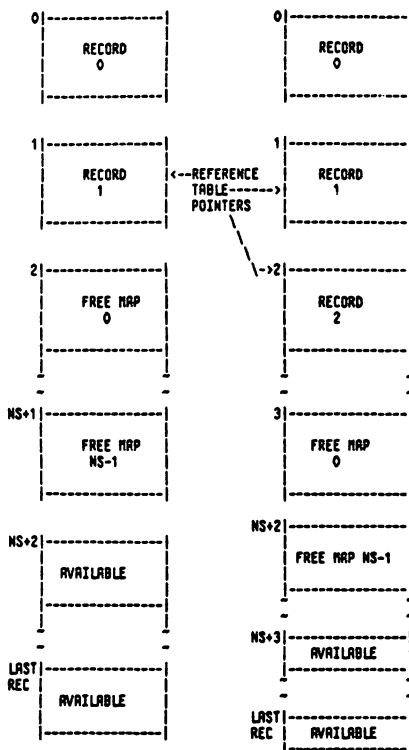
NS = Number of Sections

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 indicated that a block is used; a 1 indicated that it is free.

File space is also partitioned into 2048 record sections (16 maximum sections for Group and Account SLs, 32 maximum sections for the System SL; 2K blocks per section, one (1) map per section). The number of sections in the file is NS = (FL + 2047)/2048. The first NS records following records 0 and 1 for Group and Account SL (records 2 + NS + 1) or following records 0, 1, 2 for System SL (records 3 to NS +2) 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
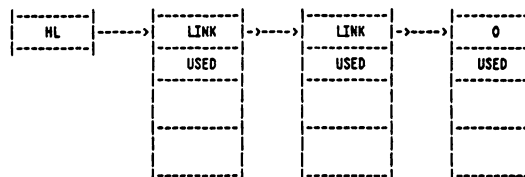
```
|----------|     |----------|   |----------|     |----------|
|   HL     |---->|   LINK   |->->|   LINK   |->--->|    0    |
|----------|     |----------|    |----------|     |----------|
                 |   USED   |    |   USED   |     |   USED  |
                 |----------|    |----------|     |----------|
                 |          |    |          |     |         |
                 |          |    |          |     |         |
                 |----------|    |----------|     |----------|
```

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 X41 TO X177.

Typical Directory Entry

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

XS- The most significant bit of segment number
    = 0 if < 256
    = 1 if >=256

NC- The number of characters in the entry point name

CN- Character count for parameters

Code Segment Linkage Structure

```
|-----------------------|
|                       |
|    CODE SEGMENT       |
|                       |
|-----------------------|
|                       |
|    STT MAP ARRAY      |
|                       |
|-----------------------|
|                       |
|    EXTERNAL LIST      |
|                       |
|-----------------------|
```
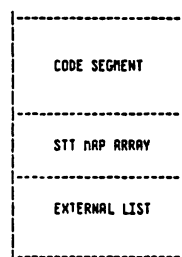
Each code segment occupies an integral number of records. This block of information can be subdivided into three tables: the CODE SEGMENT proper, an STT SEGMENT map array, and an EXTERNAL LIST.

STT Map Array

In the System SL, the STT Map Array is a 1 Word X 256 Word array. In Group and Account SLs, it is a 1 byte x 256 byte array. It is indexed by STT number. It contains the segment number which has the entry point corresponding to the external of the STT number. If no entry point in the SL matches the external, each bit in the Word (or byte) is set to one. 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.)

```
                  1 1 1 1 1 1
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
   |-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|
   |                               |
   |                               |
   |                               |
   |                               |
   |          CODE SEGMENT         |
   |                               |
   |                               |
   |                               |
   |                               |
   |-------------------------------|
   | STT MAP ARRAY                 |
   |   (SYSTEM SL=256 WORDS;       |
   |    GROUP/ACCOUNT=256 BYTES)   |
   |-|-|-|-|-------|---------------|
   |S| |X|  NC   |    CHAR (1)     |    S - SATISFIED BIT - SET IF EXTERNAL
   |-|-|-|-------|---------------|           IS SATISFIED WITHIN SL
   |          .                   |    X - 1 IF SEGN > 256
   |          .                   |    NC - NUMBER CHARACTER IN EXTERNAL
   |          .                   |    P - LEVEL OF CHECKING
   |-------------|---------------|        0 >= NONE
   |   CHAR NC   |               |        1 >= PROCEDURE TYPE
   |-------------|---------------|        2 >= # PARAMETERS
   |   STT #     |    SEG. #     |        3 >= PARAMETER TYPE
   |-|-----------|---------------|    NP - NUMBER OF PARAMETERS
   |P|   NP      |      CN       |    TN -
   |-|-----------|---------------|
   |     TN                      |
   |-----------------------------|
   |     PARM 1                  |
   |-----------------------------|
   |                             |
   |-----------------------------|
   |     PARM NP                 |
   |-----------------------------|
   |          .                  |
   |-----------------------------|
   |     0                       |    EXTERNAL LIST TERMINATOR
   |-----------------------------|
```

Reference Table Structure

Each segment has a corresponding Reference Table.

- For Group and Account segments, Reference Tables are packed four (4) entries to a record; each entry has a 32-word length. To determine entry number and displacement, divide segment number by 4: remainder = R T REC entry number, quotient = displacement into the Reference Table Pointer Map, Record 1.

- For System SL segments, Reference Tables are packed two (2) entries to a record; each entry has a 64-word length. To determine entry number and displacement, divide segment number by 2: remainder = R T REC entry number, quotient = displacement into Reference Table Pointer Map, Records 1 and 2.

When you delete a segment, the corresponding Reference Table entry is released. Free entries are linked in a list. The segment number (link) is the first word of the entry

When you add a segment, it is assigned the first free Reference Table entry number. The segment is assigned the next available Reference Table entry and space is allocated for the new entry.

Reference Table Pointer Map

```
     Group or Account                           System SL

  DREC 1        R T REC                                 R T REC
 |--------|    |--------|                          |-----------|
 |  RL    |--> |  E     |-->          |---------|  |  RL       |--> |  E     |-->
 |  0     |    |  0     |             | DREC1   |  |  0        |    |  0     |
 |--------|    |--------|             |         |  |-----------|    |--------|
 |  .     |    |  E     |             |         |  |   .       |    |  E     |
 |  .     |    |  1     |             |         |  |-----------|    |  1     |
 |--------|    |--------|             |         |  |  RL 127   |    |--------|
 |  .     |    |  E     |             |---------|  |-----------|
 |  .     |    |  2     |             | DREC2   |  |  RL 128   |
 |--------|    |--------|             |         |  |-----------|
 |  RL    |    |  E     |             |         |  |   .       |
 |  63    |    |  3     |             |         |  |-----------|
 |--------|    |--------|             |         |  |  RL 255   |
                                                   |-----------|
  (FILE REC1)  (1 SECTOR)                                       (1 SECTOR)
```

Reference Table (510 Maximum Entries)

```
              0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
             |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
         | 0| P| N|   SEGMENT LENGTH                        |0
         |  |--|--|----------------------------------------|
         | 1|   SEGMENT ADDRESS (RECORD NO.)               |1
         |  |----------------------------------------------|
         | 2|   NO. OF REC. FOR SEG AND EXTERNAL LIST      |2
         |  |--|--|----|--|--|--|-----------------------|
         | 3| F| S|    | A| C| X|   NO. ENTRY POINTS     |3   > Group
         |  |--|--|----|--|--|--|-----------------------|        and
         | 4|   STARTING ADDRESS OF PMAP                  |4   Account
         |  |--|-----------------------------------------|
         | 5|   STARTING ADDRESS OF SYMBOLIC INFORMATION  |5
         |  |--|--|-------------------------------------|
         | 6| T| X|                        FOR MPE/XL   |6
         |  |--|--|-------------------------------------|
         | 7|   SYMBOLIC INFO. LENGTH                     |7
         |  |----------------------------------------------|
         |10|   SEGMENT NAME                               |8
System<  |  |                                              |
 | SL  |17|                                              |15
         |  |----------------------------------------------|
         |20|   REFERENCE SEGMENTS                         |16
         |  |     BIT MAP (FOR BOTH SYSTEM SL              |
         |37|     AND GROUP AND ACCOUNT)                   |31
         |  |----------------------------------------------|
         |40|   EXPANDED REFERENCED                        |32
         |  |     SEGMENTS BIT MAP                         |
         |57|     (FOR SYSTEM SL ONLY)                     |47
         |  |----------------------------------------------|
         |60|                                              |48
         |  |   RESERVED                                   |
         |77|                                              |63
         |  |----------------------------------------------|
```

Where: P=1 if the code segment contains a privileged instruction.
       N=0 if the STT is in the old format.
       N=1 if the STT is in the new format.
       F=1 if the segment is deleted.
       S=1 if all external are satisfied.
       A=1 if the segment is permanently allocated.
       C=1 if the segment is core resident segment.
       X=1 if the segment is an MPE segment.
       T=1 if the segment contains a patch area.
       K=1 if the segment contains a checksum.

Code Segment With Patch Area

```
   |-------------|
   |             |
   |    CODE     |
   |             |
   |-------------|
   |             |
   | PATCH AREA  |
   |             |
   |-------------|
   |             |
   |    STT      |
   |             |
   |-------------|
```

Patch Area

```
   |-------------------|
   |   SEGMENT NAME    |     8-WORD SEGMENT 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

```
|                    1 1 1 1 1 1
|      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            |
       |------------------------------|
       |       16-BIT SEG NUM         |
       |------------------------------|
```

Type 1 Procedure PMAP Record

```
|                    1 1 1 1 1 1
|      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

```
|                    1 1 1 1 1 1
|      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 must use the MPE intrinsics or commands
which will get the load process' SIR, fill the loader communication table,
and then awaken the loader. Upon completion, the loader will return its
status through the loader communication table, and then the waiting process
will be activated.

Loader Segment Table Overview

The Loader Segment Table consists of at least two DSTs. The first one is DST
X22 and is known as the LST. The others are known as LSTX data segments and
are linked together by a doubly linked list with the list head in the LST.
SYSGLOB X226 also points to the first LSTX in the chain. The following
illustrates this structure:

```
DST X22     |---ABS (X1226)
|---------| |-->|---------|   |---------|   |---------|
|         | |-->|         |   |         |   |         |
|   LST   |<----->|  LSTX |<----->|  LSTX |<----->|  LSTX |
|         |   |         |   |         |   |         |
|---------|   |---------|   |---------|   |---------|
```
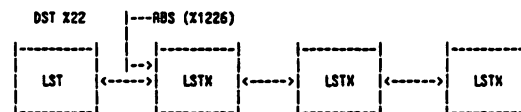
The primary purpose of the LST/LSTX data segment is to store directory
entries, which keep track of loaded code segments in program and SL files.
The LST data segment contains an additional area, known as the Loader
Communication Table, through which all processes in the system communicate
with the Loader process. Management of the messages in this area is handled
exclusively by privileged system code.

The overall layouts of the LST and LSTX data segments are as follows:

LST Overview

```
     |---------------------------|
DB|  |---------------------------|
  -  |                           |
     |      PRIMARY DB AREA       |
  -  |                           |
     |                           |
     |---------------------------|
DB+2 -->|                       |
     |  LOADER                   |
  -  |          COMMUNICATION    -
     |  TABLE                    |
     |---------------------------|
DB+7 -->| SBUF (128 WORDS)       -
  -  |                           |
     |                           |
     |---------------------------|
DB+0 -->|                       |
  -  |                           |
     |     DIRECTORY             -
     |                           |
     |                           |
   Z|  |                         |
     |---------------------------|
```

XLST Overview

```
     |---------------------------|
DB|  |---------------------------|
  -  |                           |
     |      PRIMARY DB AREA       -
  -  |                           |
     |                           |
     |---------------------------|
DB+7 -->| SBUF (128 WORDS)       |
  -  |                           -
     |                           |
     |---------------------------|
DB+0 -->|                       |
  -  |                           |
     |     DIRECTORY             -
     |                           |
   Z|  |                         |
     |---------------------------|
```
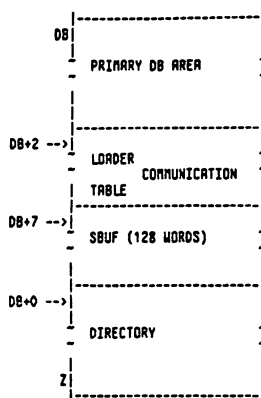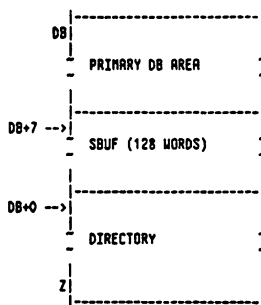
The layouts of the Primary DB areas in LST and LSTX data segments are identical. However, only those words required for management of the directory area are maintained in LSTX data segments. All other primary DB cells are defined only for the LST data segment.

The purpose of the LSTXs is to provide relatively unlimited storage for the LST directory data structure. For performance (and implementation) reasons, only those entry types related to LOADPROC functions reside in LSTX data segments. When these entries need to be accessed, they are copied into temporary entries in the directory of the LST data segment. In order to be sure that this is possible, a maximum-sized entry of each LOADPROC entry type (LOADPROCMASTER and EXTENSION) is reserved in the LST data segment by the Loader process. Pointers to these special reserved entries are stored in the Primary DB area of the LST data segment. If one of these pointers is zero, it means the Loader process was unable to allocate a maximum-sized entry, in which case a dynamically allocated entry of the required size must be allocated for the copy of the needed LSTX entry.

The first word of each permanently allocated, temporary entry is used to indicate whether or not the entry is currently in use. Zero means it's available; anything else means it's already being used to access an LSTX entry. Existing loader logic should not require access to more than one LSTX entry of a given type at a time.

In order to prevent temporary entries from accumulating in the LST data segment's directory, they must be explicitly removed (copied back to the LSTX data segment first, if modified) when no longer needed. This differs from LST-data-segment-resident entries, which require no special logic in the accessing code to release such entries. And this, in turn, is what makes moving entries from the LST to the LSTX so difficult; it's sometimes hard to tell exactly when an entry is no longer needed, and, therefore, is safe to release.

Loader Segment Table Primary DB

0	@DIR	0	24	HDFWLINK(TYPE 0)	20
1	DIR LEN	1	25	HDFWLINK(TYPE 1)	21
2	@LCT	2		.	
3	ENTP	3		.	
4	ENTP1	4	34	HDFWLINK(TYPE 8)	28
5	ENTP2	5	35	HDBKLINK(TYPE 0)	29
6	ENTP3	6	36	HDBKLINK(TYPE 1)	30
7	@SBUF	7		.	
10	SI	8	45	HDBKLINK(TYPE 8)	37
11	SJ	9	46	S/A LOADER AUX DS	38
12	SK	10	47	REFTAB DST#	39
13	SL	11	50	CUR # LSTX DSEGS	40
14	SM	12	51	MAX # LSTX DSEGS	41
15	SN	13	52	PREV LST/X DST#	42
16	SO	14	53	CURRENT LST/ X DST #	43
17	SP	15	54	NEXT LST/X DST#	44
20	SQ	16	55	TEMP EXT ENTP	45
21	SR	17	56	TEMP MASTER ENTP	46
22	SS	18	57	LCT	47
23	ST	19		.	

```
ENTPn     = Pointers point to the current accessed entry
SBUF      = Utility buffer. Usually contains program file record
            0 information
SI ~ ST   = Utility DB relative variables
SA        = Global system wide AUTOALLOCATE ON/OFF flag.
HDFWLINKs = Head of forward link for each type
HDBKLINKs = Head of backward link for each type
REFTAB DSTW= DST number for the DST which contains all the reference
            table entries for SL.PUB.SYS.
```

A reference table entry in SL.PUB.SYS is 64 words long (however, only the first 48 words are used). The data segment only contains the 48 words of the entry which are used so as to not waste space.

Directory Entries

This section shows the layouts of all LST directory entry types. The layouts given are those for permanent entries in the LST or LSTX. For temporary entries (i.e., those entries in the LST data segment which are copies of entries which reside in an LSTX data segment), the following three-word prefix should be attached:

```
 0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
|  HOME LSTX DST #                               |
|------------------------------------------------|
|  HOME LSTX OFFSET                              |
|------------------------------------------------|
|  HOME ENTRY LENGTH                             |
|------------------------------------------------|
```

ENTP in the Primary DB area always points to the word after the LENGTH word of the entry currently being accessed. ENTPn pointers are used to access other structures within the current entry. The first three words of permanent entries (or first six words of temporary entries) define the entry header words, and are negatively addressed through ENTP.

```
     0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
    |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|   GARBAGE(0)
  0 | FORWARD LINK                                   |
    |------------------------------------------------|
  1 | BACKWARD LINK                                  |
    |------------------------------------------------|
  2 | LENGTH                                         |
    |-----------------------------|------------------|
  3 |                             | 0                |
    |------------------------------------------------|
  4 | GARBAGE                                        |
    |------------------------------------------------|
```

Directory Entries (Cont.)

```
   0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
  |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|   SL FILE(1)
 0| FORWARD LINK                                    |
  |------------------------------------------------|
 1| BACKWARD LINK                                   |
  |------------------------------------------------|
 2| LENGTH                                          |
  |--|--|--|--|-------------------------------------|
 3| |XS|     |        1                             |
  |------------------------------------------------|
 4| FILE DISC ADDRESS                               |
  |-                                              - |
 5|                                                 |
  |------------------------------------------------|
 6| FILE PV INFO                                    |
  |------------------------------------------------|
 7| # ALLOCATED SEGMENTS                            |
  |------------------------------------------------|
10| # SEGLIST ENTRIES                               |
  |------------------------------------------------|
11| SEG ARRAY (32 WORD IF XS = 1)                   |
  |           (16 WORD IF XS = 0)                   |
  |------------------------------|--|--|--|--|  \
  | LOGICAL SEGMENT NUMBER       |  |A |C |X |M |   \
  |------------------------------|--|--|--|--|    \  SEGLIST ARRAY
  | REFERENCE COUNT                            |   > 3 WORD ENTRY
  |--------------------------------------------|   | PER ALLOCATED
  | PHYSICAL CST NUMBER                        |   | SL SEG
  |-                                         - |  /
  |          :                                 |
  |--------------------------------------------|
  |--------------------------------------------|
  |--------------------------------------------|
  |--------------------------------------------|
  |--------------------------------------------|
```

Where XS = 1 means entry is for the Extended System SL bit map
          of 32 words.
      A = 1 means code segment is allocated.
      C = 1 means code segment is core resident.
      X = 1 means code segment is an MPE system segment.
      M = 1 means code segment is physically mapped.
        = 0 means code segment is logically mapped.

Directory Entries (Cont.)

```
   0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
  |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|   PROGRAM
 0| FORWARD LINK                                    |  FILE (2)
  |------------------------------------------------|
 1| BACKWARD LINK                                   |
  |------------------------------------------------|
 2| LENGTH                                          |
  |--|--|--|--|------------|-------------------------|
 3|P |A |XS|AR|    LIB     |        2               |
  |------------------------------------------------|
 4| FILE DISC ADDRESS                               |
  |-                                              - |
 5|                                                 |
  |------------------------------------------------|
 6| CST BLOCK INDEX                                 |
  |------------------------------------------------|
 7| SEGMAP DST                                      |
  |------------------------------------------------|
10| # PROCESS SHARING                               |
  |------------------------------------------------|
11| # SEG IN PROGRAM    | # SLINFO AREA             |
  | FILE                |                           |
  |------------------------------------------------|
12| PV FILE INFO                                    |
  |------------------------------------------------|
13| TRACE EXTERNAL PLABEL                           |  \
  |--|----------------------------------------------|  |
14|S | SL SEARCH SEQUENCE                           |  |
  |------------------------------------------------|  |
15| SL FILE DISC ADDRESS                            |  | SL INFO AREA
  |-                                              - |  |
16|                                                 |  > 35 WORDS FOR
  |------------------------------------------------|  | SYSTEM SL
17|                                                 |  | 19 WORDS FOR
  |- LIB SEG ARRAY (16 WORDS)                     - |  | GROUP AND
  |------------------------------------------------|  | ACCOUNT SL
  |          :                                      |  /
  |------------------------------------------------|
  | PSEGMAP SIZE                                    |  \
  |-----------------------------|------------------|  |
  | LIB LOG SEG (9 BITS)        | SL INFO INDEX    |  |
  |-----------------------------|------------------|  | PSEGMAP
  | LIB LOG SEG (9 BITS)        | SL INFO INDEX    |  > ARRAY
  |-----------------------------|------------------|  |
  |          :                                      |  |
  |-----------------------------|------------------|  |
  | LIB LOG SEG (9 BITS)        | SL INFO INDEX    |  |
  |-----------------------------|------------------|  /
```

Directory Entries (Cont.)

Where P = 1 means program is executed with NOPRIV option.
      A = 1 and AR = 0 means program is allocated.
      AR = 1 and A = 1 means program is auto allocated.
      XS = 1 means entry contains an Extended System SL bit map
           of 32 words.
      S = 1 means this bit map is for the Extended System SL and
          contains 32 words.

```
   0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
  |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|   LOADING(3)
 0| FORWARD LINK                                    |
  |------------------------------------------------|
 1| BACKWARD LINK                                   |
  |------------------------------------------------|
 2| LENGTH                                          |
  |--|----------------------|-----------------------|
 3|P |                      |        3              |
  |------------------------------------------------|
 4| FILE DISC ADDRESS                               |
  |-                                              - |
 5|                                                 |
  |------------------------------------------------|
```

```
   0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
  |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|   WAITER(4)
 0| FORWARD LINK                                    |
  |------------------------------------------------|
 1| BACKWARD LINK                                   |
  |------------------------------------------------|
 2| LENGTH                                          |
  |--|----------------------|-----------------------|
 3|P |                      |        4              |
  |------------------------------------------------|
 4| FILE DISC ADDRESS                               |
  |-                                              - |
 5|                                                 |
  |------------------------------------------------|
 6| WAITING PIN                                     |
  |------------------------------------------------|
 7| UNUSED                                          |
  |------------------------------------------------|
```

Directory Entries (Cont.)

```
   0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
  |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|   LOADED(5)
 0| FORWARD LINK                                    |
  |------------------------------------------------|
 1| BACKWARD LINK                                   |
  |------------------------------------------------|
 2| LENGTH                                          |
  |--|----------------------|-----------------------|
 3|P |                      |        5              |
  |------------------------------------------------|
 4| FILE DISC ADDRESS                               |
  |-                                              - |
 5|                                                 |
  |------------------------------------------------|
 6| LOAD PROCESS STATUS                             |
  |-                                              - |
 7|                                                 |
  |------------------------------------------------|
```

```
   0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
  |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|   SHARER(6)
 0| FORWARD LINK                                    |
  |------------------------------------------------|
 1| BACKWARD LINK                                   |
  |------------------------------------------------|
 2| LENGTH                                          |
  |--|----------------------|-----------------------|
 3|P |                      |        6              |
  |------------------------------------------------|
 4| PIN                                             |
  |------------------------------------------------|
 5| FILE DISC ADDRESS                               |
  |-                                              - |
 6|                                                 |
  |------------------------------------------------|
```

Directory Entries (Cont.)

```
 |     0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
 |    |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|    EXTENSION(7)
 |  0| FORWARD LINK                              |
 |    |------------------------------------------|
 |  1| BACKWARD LINK                             |
 |    |------------------------------------------|
 |  2| LENGTH                                    |
 |    |-----|--|--|------------------------------|
 |  3| XS|SD| LIB |        7                     |
 |    |-----|--|--|------------------------------|
 |  4| PIN                                       |         Word contains
 |    |------------------------------------------|         LOADPROC count,
 |  5| EXTENSION ID                              |         if entry was
 |    |------------------------------------------|      \  created by
 |  6| LOADPROC COUNT(LOADPROC) /                |      || LOADPROC.
 |    | LOG SEG # (ALLOCATEPROC)                 |      /  Word contains
 |    |------------------------------------------|         logical segment
 |  7| PLABEL                                    |         number, if entry
 |    |------------------------------------------|         was created by
 | 10| # CHAR IN NAME   |                        |         ALLOCATEPROC.
 |    |------------------|                        |
 | ~  PROCEDURE NAME                             ~
 |    |------------------------------------------|
 |    |                  | # SL INFO AREA        |
 |    |--|---------------|-----------------------|  \
 |  S | SL SEARCH SEQUENCE                        |  | SL INFO AREA
 |    |------------------------------------------|  | 35 WORDS FOR
 |    | SL FILE DISC ADDRESS                      |  | SYSTEM SL
 |    |------------------------------------------|  | 19 WORDS FOR
 |    | LIB SEG ARRAY (32 WORD IF S = 1)          |  | GROUP AND
 |    |                (16 WORD IF S = 0)         |  | ACCOUNT SL
 |    |------------------------------------------|  /
 |    | MCSTREFSIZE                               |
 |    |--|---------------|-----------------------|  \
 |  N |                  | MCST INDEX (1)         |  |
 |    |--|---------------|-----------------------|  |
 |    | ~       .                                ~  > MCSTREF ARRAY
 |    |--|---------------|-----------------------|  |
 |  N |                  | MCST INDEX(n)          |  |
 |    |--|---------------|-----------------------|  /
```

XS = 1 means entry contains an Extended System SL bit map
    of 32 words.
SD = Search Domain
    = 1 means use program group and account for library search.
    = 0 means use log on group and account for library search.
S = 1 means this bit map is for the Extended System SL and
    contains 32 words.

Directory Entries (Cont.)

```
 |     0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
 |    |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|   LOADPROC
 |  0| FORWARD LINK                              |      MASTER(8)
 |    |------------------------------------------|
 |  1| BACKWARD LINK                             |
 |    |------------------------------------------|
 |  2| LENGTH                                    |
 |    |-------------------|----------------------|
 |  3|                    |        8             |
 |    |-------------------|----------------------|
 |  4| PIN                                       |
 |    |------------------------------------------|
 |  5| # SLID ENTRIES     | # ACTIVE LOADPROCS   |
 |    |------------------------------------------|
 |  6|                                           |
 | ~  EXT IDX TABLE (16 WORDS)                   ~
 | 25|                                           |
 |    |------------------------------------------|
 | 26|                                           |
 | ~  MCST IDX TABLE (16 WORDS)                  ~
 | 45|                                           |
 |    |------------------------------------------|  \
 | 46| SLID(1)                                   |  |
 |    | DISC ADDRESS OF SL                        |  | REFERENCED
 |    |------------------------------------------|  > SL ARRAY
 |    | ~     .                                  ~  | 2 WORDS
 |    | ~     .                                  ~  | PER ENTRY
 |    |------------------------------------------|  |
 |    | SLID(n)            |                     |  |
 |    |--------------------|---------------------|  /
 |    |                    | # MCST LOGSEG SIZE  |
 |    |--------------------|---------------------|
 |    | LIB LOG SEG (9 BITS) | SL INFO INDEX     |  \
 |    |----------------------|-------------------|  |
 |    | REFERENCE COUNT                           |  |
 |    |------------------------------------------|  | MCST LOGSEG
 |    | ~     .                                  ~  > ARRAY
 |    | ~     .                                  ~  | 2 WORDS PER
 |    |------------------------------------------|  | ENTRY
 |    | LOG SEG #          | SLID INDEX(n)       |  |
 |    |--------------------|---------------------|  |
 |    | REFERENCE COUNT                           |  |
 |    |------------------------------------------|  /
```

Loader Cache

SYSGLOB extension area + X72 contains DST number of cache
BUCKETSIZE = X52

Cache Data Segment Format

```
                  |----------------------|
              0|  |                      |
              1|  | HIT COUNTER          |
                  |----------------------|
              2|  |                      |
              3|  | MISS COUNTER         |
                  |----------------------|
              4|  | BUCKET 0             |
                  |----------------------|
              | BUCKET 1             |
   4+ BUCKETSIZE  |----------------------|
                  | ~    .            ~   |
                  |----------------------|
 4+94ª BUCKETSIZE |                      |
                  | BUCKET 94            |
 4+95ª BUCKETSIZE -1 |-------------------|
```

Bucket Format

```
   |-------------------|
 0| LENGTH OF         |
  | SLDIR1 + 1        |
   |-------------------|
 1| SLDIR 1           |   MOST RECENTLY REFERENCED SYSTEM SL
   |-------------------|   DIRECTORY ENTRY FROM THIS SL DIRECTORY
  | LENGTH OF         |   BUCKET
  | SLDIR2 + 1        |
   |-------------------|
  | SLDIR 2           |   SECOND MOST RECENTLY REFERENCED ENTRY
   |-------------------|
  | ~     .        ~   |
   |-------------------|
  | LENGTH OF         |
  | SLDIRN + 1        |
   |-------------------|   Nth MOST RECENTLY REFERENCED ENTRY; IF
 BUCKET| SLDIRN        |   NOT COMPLETE THEN INDICATES END OF
 SIZE-1|----------------|  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
 |  4|                                           |          0 = load prgm
 |    |                                           |          1 = load proc
 |  5|                                           |          2 = alloc prog
 |    |                                           |          3 = alloc proc
 |  6|                                           |        LIB = library search
 |    |                                           |          0 = SYS
 |  7| UNUSED                                    |          1 = PUB
 |    |                                           |          2 = GROUP
 | 10|                                           |        M = NON PRIV MODE
 |    |                                           |        LD = LOAD DOMAIN (FOR
 | 11|                                           |             LOADPROC)
 |    |                                           |          0 = USE LOGON GRP
 | 12|                                           |             & ACCT SLs
 |    |------------------------------------------|          1 = USE GROUP &
 | 13| WRITER PCB INDEX                          |             ACCT SLs WHERE
 |    |--|--|--|--|-----|--|--|--|--|             |             THE PROGRAM
 | 14|       |BA|IA|PM|     |MR|   |DS|PH|        |             RESIDES
 |    |--|--|--|--|-----|--|--|--|--|             |        L = USER CAPABILITY
 | 15|                                           |
 |    |------------------------------------------|
 | 16| GROUP                                     |
 |    |------------------------------------------|
 | 17| NAME                                      |
 |    |------------------------------------------|
 | 20|                                           |
 |    |------------------------------------------|
 | 21|                                           |
 |    |------------------------------------------|
 | 22| ACCOUNT                                   |
 |    |------------------------------------------|
 | 23| NAME                                      |
 |    |------------------------------------------|
 | 24|                                           |
 |    |------------------------------------------|
 | 25| DV INFO                                   |
 |    |------------------------------------------|
```

Loader Communication Table (LCT) (Cont.)

Form Incoming to Loader (Load/Allocate Procedure)

```
    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| EXTENSION ID           |
   |------------------------|
 3| # CHAR IN NAME         |
   |------------------------|
 4|                        |
   |                        |
 5|                        |
   |                        |
 6| PROCEDURE NAME         |
   |                        |
 7|                        |
   |                        |
10|                        |
   |                        |
11|                        |
   |                        |
12|                        |
   |------------------------|
13| WRITER PCB INDEX       |
   |------------|--|--|--|--|----|--|--|--|
14|            |BA|IA|PM|    |MR| |DS|PH|  USER CAPABILITY
   |------------|--|--|--|--|----|--|--|--|
15|                        |
   |------------------------|
16| GROUP                  |
17| NAME                   |
20|                        |
   |------------------------|
21|                        |
22| ACCOUNT                |
23| NAME                   |
24|                        |
   |------------------------|
25| PV INFO                |
   |------------------------|
```

CMD = loader cmd
    0 = load prgm
    1 = load proc
    2 = alloc prog
    3 = alloc proc
LIB = library search
    0 = SYS
    1 = PUB
    2 = GROUP

M = NONPRIV MODE
LD = LOAD DOMAIN
L = LOAD MAP REQ.

Loader Communication Table (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(X17). 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 (X221 and X222) 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          | v
  |   |------------------------------|
  |   | PHYSICAL SEGMENT #           | ^
  |   |------------------------------| | LOGICAL SEG 2
  |   | POINTER TO STT LIST          | v
  |   |------------------------------|
  |   |         .                    | ^
  |   |         .                    | .
  |   |         .                    | v
  |   |------------------------------|
  |   | PHYSICAL SEGMENT #           | ^
  |   |------------------------------| | LOGICAL SEG n
 /-|--| POINTER TO STT LIST          | v (MAX 255)
 | |  |-----|------|------|----------|
 | |  | M| STT # | SEG # |          | ^
 | |  |-----|------|------|          | STT'S FOR LOGICAL
 | |  | M| STT # | SEG # |          | v SEGMENT 1
 | |  |-----|------|------|          |   (IF NEEDED)
 | |  |         .                    |
 | |  |-----|------|------|          |
 | |  | M| STT # | SEG # |          |
 | |  |-----|------|------|          |
 | \->| TOTAL STT'S FOR THIS SEG     |
 |    |------------------------------|
 |    |         .                    | ^
 |    |         .                    | .
 |    |         .                    | v
 |    |-----|------|------|----------|
 |    | M| STT # | SEG # |          | ^
 |    |-----|------|------|          | STT'S FOR LOGICAL
 |    | M| STT # | SEG # |          | SEGMENT n
 |    |-----|------|------|          | (IF NEEDED)
 |    |         .                    |
 |    |-----|------|------|          |
 |    | M| STT # | SEG # |          |
 |    |-----|------|------|          |
 \--->| TOTAL SST'S FOR THIS SEG     | v
      |------------------------------|
```

Loader Auxiliary Data Segment

Overview

The loader auxiliary data segment is a multi-table data segment. Each table
within the data segment has a standard header and they are linked together
via DB relative pointers.

Tables within the loader auxiliary data segment are the:

1. Autoallocate table
2. Program name table.

Loader Auxiliary Data Segment Format

```
          /  0 |------------------------------| \
          |    | TABLE SIZE                   | |
          |  1 |------------------------------| |
          |    | ENTRY LENGTH                 | |
          |  2 |------------------------------| |
          |    | TOTAL NUMBER FREE            | |
          |  3 |------------------------------| |
          |    | FIRST FREE                   | |
  Table --|  4 |------------------------------| |-- Table one
  Header  |    | DB RELATIVE POINTER TO TABLE TWO | |
          |  5 |------------------------------| |
          |    | UNUSED                       | |
          |  6 |------------------------------| |
          |    | UNUSED                       | |
          \  7 |------------------------------| /
               | UNUSED                       |
               |------------------------------|
               |                              |
               | AUTOALLOCATE TABLE           |
               |                              |
               |------------------------------|
          /  0 |------------------------------| \
          |    | TABLE SIZE                   | |
          |  1 |------------------------------| |
          |    | ENTRY LENGTH                 | |
          |  2 |------------------------------| |
          |    | UNUSED                       | |
  Table --|  3 |------------------------------| |
  Header  |    | UNUSED                       | |
          |  4 |------------------------------| |
               | DB RELATIVE POINTER TO NEXT TABLE | |
               |------------------------------|
```

Loader Auxiliary Data Segment Format (Cont.)

```
    |  |-----------------------------|  |-- Table two
    | 5|  UNUSED                     |  |
    |  |-----------------------------|  |
    | 6|  UNUSED                     |  |
    |  |-----------------------------|  |
   \ 7|  UNUSED                     |  |
    |  |-----------------------------|  |
    |  |                             |  |
    |  |                             |  |
    |  |                             |  |
    |  |   PROGRAM NAME TABLE        |  |
    |  |                             |  |
    |  |                             |  |
    |  |-----------------------------|  /
```

Autoallocate Table

Overview

The autoallocate table is a table to aid in the automatic allocation/
deallocation of programs. It is a least recently used list of programs which
are not being used at the moment and thus are candidates to be automatically
deallocated.

Entry Contents

Each entry is a DB relative address of a program file entry in the loader
segment table (LST).

Autoallocate Table

```
    |-----------------------------|
  0 |  LEAST RECENTLY USED ENTRY  |
    |-----------------------------|
    |                             |
    |              .              |
    |              .              |
    |              .              |
    |-----------------------------|
    |                             |
    |  MOST RECENTLY USED ENTRY   |
    |-----------------------------|
```

Program Name Table

Overview

The program name table contains the fully qualified file names and the disc
address of all programs currently loaded in the system. The name of a
program can be found by indexing into the program name table with the same
index used for the CSTX block table. This index can be found in the PCB and
in the program entry of the LST.

When a program is loaded, its name is entered into this program name table.
However, when it is unloaded, it is not removed. Thus the program name table
entry "x" will contain the name of the program which is currently allocated
to the corresponding CSTX block index. If the corresponding CSTX block
index is free then the contents of the program name table is the program
which was last allocated to that index.

NOTE: Entry 0 of the program name table is used by the loader as a
      scratch area since there is no legal CSTX block index of zero.

```
         |-----------------------------|
  |    0 |                             | 0
  |      |                             |
  |    1 |  PROGRAM                    | 1
  |      |                             |
  |    2 |  NAME                       | 2
  |      |                             |
  |    3 |                             | 3
  |      |-----------------------------|
  |    4 |                             | 4
  |      |                             |
  |    5 |  PROGRAM                    | 5
  |      |                             |
  |    6 |  GROUP                      | 6
  |      |                             |
  |    7 |                             | 7
  |      |-----------------------------|
  |   10 |                             | 8
  |      |                             |
  |   11 |  PROGRAM                    | 9
  |      |                             |
  |   12 |  ACCOUNT                    | 10
  |      |                             |
  |   13 |                             | 11
  |      |-----------------------------|
  |   14 |  FILE                       | 12
  |      |  DISC                       |
  |   15 |  ADDRESS                    | 13
  |      |-----------------------------|
  |   16 |  UNUSED                     | 14
  |      |-----------------------------|
  |   17 |                             | 15
  |      |-----------------------------|
```

CHAPTER 12 PRIVATE VOLUMES / SERIAL DISC

Mounted Volume Table (MVTAB)

```
                              DST = 53 = X65
                              SIR = 27 = X33

                        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)
  - |            .                | -              |
    |            .                |                 |
    |                             |                 |
 21 |  0                          | 17             |
    |-----------------------------|                 |
 22 |  0                .         | 18             |
    |-----------------------------|                 |
 23 |  0                          | 19             |
    |-----------------------------|                 |
 24 |  0                          | 20             v
    |-----------------------------|                ---
```

Mounted Volume Table (MVTAB) (Cont.)

```
    |-|-----|-------------------|
  0 |U| CYCL|   DIRCSIZE/32     | 0           ---
    |-|-----|-------------------|              ^
  1 |  HVOL |  HVOL             | 1           |
    |-------|-------------------|              |
  2 |  LDEV |    DIRBASE        | 2   MASTER VOLUME
    |---------------------------|     OF VOLUME SET
  3 |  OF VOLUME SET            | 3   IS ON THIS LDEV
    |---------------------------|              |
  4 |  UCNT                    | 4           |
    |---------------------------|              |
  5 |  GENERATION NUMBER        | 5           \
    |-------|-------------------|              |
  6 |  LDEV |   VTABX           | 6  \        |
    |-------|-------------------|     |       | > - VOL ENTRY 0  > ENTRY 1
  7 |  VCNT                    | 7  |       |   (DOUBLE)       (MVTABX = 1)
    |---------------------------|     /       |
  - |            .              | -           |
  - |            .              | -           |
    |-------|-------------------|              |
 24 |  LDEV |   VTABX           | 20 \        |
    |-------|-------------------|     | > - VOL ENTRY 7
 25 |  VCNT                    | 21 |   (DOUBLE)       v
    |---------------------------|     /              ---
    |                           |                     ^
  - |                           | -                   |
    |                           |                     |
    |                           |                     | > ENTRY n - 1
  - |                           | -                   |   (MVTABX = n - 1)
    |                           |                     |
    |                           |                     v
    |---------------------------|                    ---
```

Mounted Volume Table (MVTAB) (Cont.)

```
  |-|-|----|-----------------|
 0|U|CYCL| DIRCSIZE/32     |0
  |-|-|----|-----------------|
 1| MVOL | NVOL           |1
  |------|-----------------|
 2| LDEV    |  DIRBASE     |2
  |---------|--------------|
 3|   OF VOLUME SET        |3
  |-----------------------|
 4|   UCNT                 |4
  |-----------------------|
 5|   GENERATION NUMBER    |5          \
  |------|-----------------|            \
 6| LDEV |  VTABX          |6  |         > - VOL ENTRY 0 > - ENTRY n
  |------|-----------------|   |            (DOUBLE)       (MVTABX = n)
 7|   VCNT                 |7  /
  |-----------------------|
  |          .            |
  -          .            -
  |                       |
  |------|-----------------|  \
24| LDEV |  VTABX          |20  \
  |------|-----------------|    > - VOL ENTRY 7
25|   VCNT                 |21  /    (DOUBLE)
  |-----------------------|
```

    U   - In Use (1)
           Not Used (0)

    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

    VCNT - # of users having mounted
           the volume.

Private Volume User Table (PVUSER)

                        DST = 54 (X66)/SIR = 29 (X35)

```
                                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         |2                |
      |   REPRESENTED              |                 > - TABLE HEAD
  3|   MAXIMUM TABLE SIZE (WORDS)  |3                |   (5 WORDS)
      |-----------------------------|                |
  4|   AVAILABLE POINTER           |4                v
      |--------------|--------------|              ---
  5|   OP MASK       |   MVTABX     |5  \            ^
      |--------------|--------------|    \           |
  6|   MAX USERS                   |6   |            |
      |-----------------------------|    |           |
  7|   # PINs                      |7    > - ENTRY HEAD
      |-----------------------------|    |   (5 WORDS)
 10|   CURRENT SIZE OF ENTRY       |8   |            |
      |-----------------------------|    |           |
 11|   PV FLAGS        |OP        |9   /            |
      |------------------|----------|               |
 12|   VMASK                       |10 \            |
      |-----------------------------|    \          |
 13|   PIN                         |11   |          |
      |-----------------------------|    |          |
 14|   USER BIND COUNT             |12   |          |
      |-----------------------------|    |          |
 15|   USER MOUNT COUNT            |13   |          |
      |-----------------------------|    |          |
 16|   SYSTEM BIND COUNT           |14   > - USER ENTRY
      |-----------------------------|    |    1
 17|   SYSTEM MOUNT COUNT          |15   |          |
      |-----------------------------|    |          |
 20|   BIND NAMES COUNT            |16   |          |
      |-----------------------------|    |          |
 21|   DST # OF BIND NAMES         |17   |          |
      |   SEGMENT                  |    /           |
      |-----------------------------|                |
 22|   VMASK                       |18 \            |
      |-----------------------------|    \          |
 23|   PIN                         |19   |          > - VOLUME SET
      |-----------------------------|    |          |   ENTRY 1
 24|   USER BIND COUNT             |20   |          |   (MVTABX = J)
      |-----------------------------|    |          |
 25|   USER MOUNT COUNT            |21   |          |
      |-----------------------------|                |
```

Private Volume User Table (PVUSER) (Cont.)

```
      |-----------------------------|
 26|   SYSTEM BIND COUNT           |22  > - USER ENTRY
      |-----------------------------|    |    2
 27|   SYSTEM MOUNT COUNT          |23  |
      |-----------------------------|    |
 30|   BIND NAMES COUNT            |24  |
      |-----------------------------|    |
 31|   DST # OF BIND NAMES         |25  |
 32|   SEGMENT                     |26  /
      -             .             -
      |             .             |
      |-----------------------------|  \
      |   VMASK                    |    \
      |-----------------------------|    |
      |   PIN                      |    |
      |-----------------------------|    |
      |   USER BIND COUNT          |    |
      |-----------------------------|    |
      |   USER MOUNT COUNT         |    |
      |-----------------------------|    > - USER ENTRY
      |   SYSTEM BIND COUNT        |    |    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
      |                             |                |
      -                             -                |
      |                             |                |
      |-----------------------------|                |
 10|   0                           |8                v
      |-----------------------------|              ---
  0|   BIND COUNT                  |0                ^
      |-----------------------------|                |
  1|                               |1                |
      -                             -                |
  2|-                             -|2                |
  3|-   GROUP NAME                -|3                |
  4|-                             -|4                > - ENTRY 1
      |-----------------------------|                |
  5|-                             -|5                |
  6|-   ACCOUNT NAME              -|6                |
  7|-                             -|7                |
      |-----------------------------|                |
 10|                               |8                v
      |-----------------------------|              ---
      |                             |
      |                             |
      |-----------------------------|
```

Bind Names Data Segment (Cont.)

```
 -                          -        ---
|                          |          ^
|--------------------------|          |
0|     BIND COUNT           |0         |
1|-                        -|1         |
2|-                        -|2         |
 |     GROUP NAME           |          |
3|-                        -|3         |
 |-                        -|4         > - ENTRY n
4|--------------------------|          |
5|-                        -|5         |
6|-                        -|6         |
 |     ACCOUNT NAME         |          |
7|-                        -|7         |
10|-                       -|8         v
 |--------------------------|         ---
|             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 cartridge tape (CTAPE). To accomplish
this, data is buffered within SDISC. The buffer is an integral number of
sectors (blocks for the CTAPE) 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 |  .....              |
|----|----|----|--------------------|
```
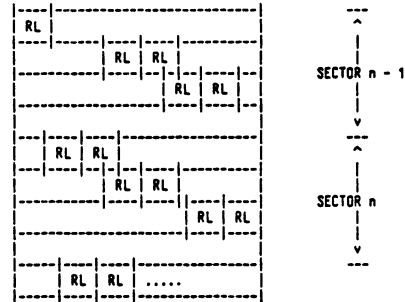
The reason for the trailing byte count is to implement an easy way to
backspace records.

End-of-File Format

Files start on a sector boundary, and end on one. The End-of-File consists
of a 0 record length and 0-fill to the end of the current sector as follows:

```
|------------------------------|
|                      RL RL   |
|                     |--------|
|  RL RL              |        |      SECTOR n
|             |-------------|  |
|          RL | 0          |  |
|-------------|            |  |
|  ZERO FILL               |  |
|------------------------------|
```
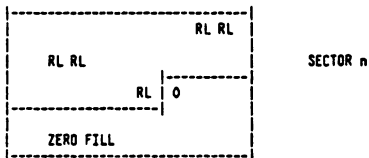
An End-of-File entry is made in the Gap Table, so that files may be skipped
by scanning Gap Table entries instead of serially scanning the data area.
Refer to "Gap Table Format" in this chapter for detailed information.

Contiguous Block Format

A serial disc can perform all the tasks that a magnetic tape can do. It can
also be a coldload device. The 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 interpret the record
length words which surround standard data records. A structure, called a
CONTIGUOUS BLOCK, which has the data without the length words is utilized.
Information as to the length of each contiguous block is stored elsewhere.
Entries in the Gap Table hold the beginning and ending sector addresses of
each contiguous block. Each block must begin and end on a sector boundary.
To distinguish contiguous blocks from normal data, a record length and a fill
character of X177777 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
|------------------------------|        ---
```
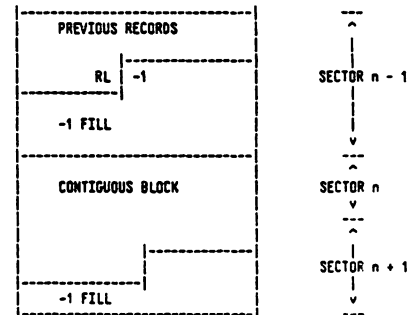
Hole Format

Holes on the serial disc have the same format as contiguous blocks (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 is on the device, starting in sector 4, while a working copy is in main memory. The copy in memory is posted to the disc only when a backspace or rewind operation occurs after writing (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 CTAPE blocks)
HP 7920	44
HP 7925	106
HP 7933/35	219 (250 for G.00.00 and later releases.)
HP 7902/9895	26
HP 35401/HP 9110/HP 9144	4 blocks ("S" cartridge)
HP 35401/HP 9110/HP 9144	15 blocks ("L" cartridge)

The following is an illustration of the Gap Table:

```
                      1 1 1 1 1 1
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
    |-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-| \
   0| SECTOR ADDR OF LOAD POINT  | |  |
   1|            UNUSED          | |  |
   2|            UNUSED          | |  > - GAP TABLE HEADER
   3|            UNUSED          | |  |
    |-----|---------------------------| /
   4| TYPE|                       |
    |-----|    SECTOR ADDRESS     | ENTRY (2 WORDS)
   5|
    |-----|---------------------------|
   6| TYPE|                       |
    |-----|    SECTOR ADDRESS     | ENTRY (2 WORDS)
   7|
    |---------------------------------|
                   .
                   .
                   .
```
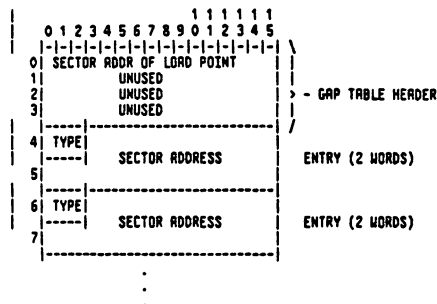
The type field is bits 0, 1 and 2 of the first word. The eight possible types are:

0. END-OF-FILE. The associated sector address contains one or more end-of-file fill characters (0) to fill out that sector. If the record ends exactly at a sector boundary, the following end-of-file sector contains all zeros.

1. END-OF-DATA. The associated sector address is the last address of valid data plus 1, (the next available address). Such an entry is usually preceded by an end-of-file entry. The EOD entry is written when writing terminates. The file system will not 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. It is usually found 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. It is always found 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 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 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 the INITIAL serial disc interface. When read by the MPE SDISC, it is 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. This type is created whenever the Gap Table is cleared, by initializing the table to -1.

SDISC Extra Data Segments

With few exceptions, SDISC operates entirely in split-stack mode using an extra data segment for 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, the 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 releases the XDS in processing the final FCLOSE against the device. The system program PVPROC may also acquire and release an XDS to allow the tape label routines in LABSEG to use SDISC when DEVREC processes a device on-line interrupt. SDISC allocates the two data buffer segments as they are needed, then releases them as part of the Device Close procedure.

The XDS contains the global storage area for SDISC, including the data buffer management area (BUFFER'INFO), and a small buffer (called WORKTABLE). The contents of the Serial Disc label sector is stored in WORKTABLE when it is read in by SDISC as part of the self-configuration. The Defective Tracks Table (MAC family discs) or Defective Sector Table (CS80 discs) is also stored in WORKTABLE while suspect or deleted tracks are being reassigned.

The three arrays in the XDS (WORKTABLE, BUFFER'INFO and GPT (Gap Table)) are dynamically configured by SDISC as indirect arrays. The array names are declared as pointers, then appropriately computed element-0 addresses are inserted in them.

The extra data segment is organized as follows:

```
    |--------------------|
  0 |   WORDSPERSECTR    | 0
    |--------------------|
  1 |   SECTORSPERTRAK   | 1
    |--------------------|
  2 |   STARTADDRESS     | 2 SIMULATED BEGINNING-OF-TAPE
    |--------------------|
  3 |   EOTSECTR         | 3 SIMULATED END-OF-TAPE
    |--------------------|
  4 |                    | 4
    |--------------------|
  5 |   EODSECTR         | 5 LAST SECTOR OF DISC
    |--------------------|
  6 |                    | 6
    |--------------------|
  7 |   JUSTALLOCATED    | 7
    |--------------------|
 10 |   WRITERING        | 8
    |--------------------|
 11 |   FATALERROR       | 9
    |--------------------|
 12 |   VOLUME'FATAL     | 10
    |--------------------|
 13 | NON'VOL'SPECIFIC'FLAGS | 11
    |--------------------|
 14 |   MAX'DSEG'SIZE    | 12
    |--------------------|
 15 |   SDISC            | 13
    |--------------------|
    |                    |
    |     WORKTABLE      |
    |                    |
    |--------------------|
    |    BUFFER'INFO     |
    |--------------------|
    |     GAP TABLE      |
    |                    |
    |--------------------|
```

The first thirteen words are reserved for use when the data segment is created. The first seven words are filled with information taken from the label sector, and the last six are filled by ALLOCATE.

WORDSPERSECTR — Words per sector

SECTORSPERTRAK — Sectors per track

STARTADDRESS — Simulates Beginning-of-Tape

EOTSECTR — Simulates End-of-Tape

EODSECTR — Simulates tape runoff

JUSTALLOCATED — Initializes SDISC parameters to BOT if true

WRITERING — Simulates tape write ring

FATALERROR — Disables SDISC permanently when true

VOLUME'FATAL — Disables SDISC until a new volume is mounted when true

NON'VOL'SPECIFIC'FLAGS — SDISC global flags that are non-volume specific.

MAX'DSEG'SIZE — Maximum size of the XDS

SDISC — Global variables, including array pointers

WORKTABLE — Length is 512 words

BUFFER'INFO — Length is calculated as:
MAX'NUM'BUFFERS (currently 2)
INFO'ENTRY'SIZE (currently 8)

GAP TABLE — Length varies with device — is calculated by the SDISC routine as part of self-configuration

Serial Disc Organization

The disc is organized as follows:

```
|------------------------|
|   LABEL SECTOR         |0   See expanded view in Chapter 3.
|--  ------------------- |
|   DTT/DSC'             |1   DTT (MAC family) or DSCT (CS80).
|------------------------|
|   COLD LOAD            |2   HP-IB cold load channel prog.
|------------------------|
|   SOFTDUMP             |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
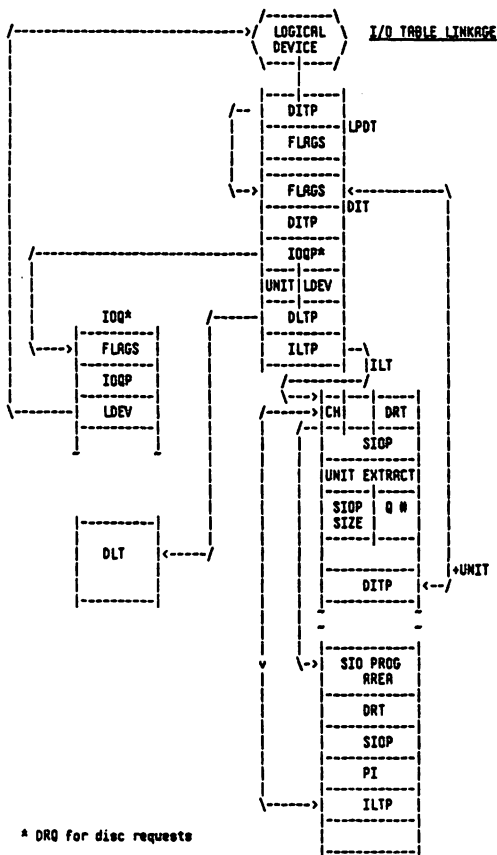
* DRQ for disc requests

Device Reference Table (DRT)

```
           HP-IB SYSTEMS
     ABS|------------------------|
  |  %10|    BANK OF DRT         |  \
                                    >------\
  |  %11|  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	
0	QUEUE NUMBER	DF MC CR (SEE BELOW) 0 MTYP	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 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 - 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 (X15)
| SIR = 9 (X11)

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 INPs).

A given virtual device entry is in use only while the devicefile it represents is open. When the file is closed by FCLOSE, the entry becomes available for another virtual device. This is the reason for the SYSDUMP/INITIAL configurator question MAX N 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|                                              |
 |----------------------------------------------|
```

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 using DISABLE/ENABLE, 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|                                           |
 |--|--|--|--|--|--|--------|--|--|--|----------|
1| 0|  | 2| 3|  | 5| 6| 7 (EOF)|  |  | 12 SUBTYPE|
 |--|--|--|--|--|--|--|--------|--|--|--|-------|
2|     SYSDB - RELATIVE POINTER TO THE DIT      |
 |----------------------------------------------|
3|                                              |
 |----------------------------------------------|
```

0 - 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 - Device reserved (alternate use) - set during :STARTSPOOL to remove the device from the pool of available devices while other checks are made or resources are acquired; the field is set to 1 when these steps are completed.

2 - Word 1.(2:1) - Device is Job/Session Accepting if true.

3 - Word 1.(3:1) - Device is Data Accepting if true.

5 - Word 1.(5:1) - Device is Duplicative if true (all devices except discs).

6 - Word 1.(6:1) - Device is Interactive if true (all devices except discs).

7 - 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.

12 - 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|                                              |
  |--|--|--|--|--|--|--------|--|--|------------|
1|   |   |   | 4 |   |   |        |10|11|            |
  |-----|---|---|---|---|--------|--|--|------------|
2|     SYSDB - RELATIVE POINTER TO THE DIT         |
  |-----------------------------------------------|
3|       | 3|                                       |
  |-----|--|--------------------------------------|
```

3 - Word 3.(3:1) - If set, NLIO translation is invoked for ALL data
transferred to and from the device.

4 - Word 1.(4:1) - Control-Y is allowed and has been detected.

10 - Word 1.(10:1) - BREAK has been detected. Ignore BREAK if the CI is
running.

11 - 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. Only IOTERMO and HIOTERM 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|                                              |
  |--|--|--|--|--|--|--|--------|--|--|--|---------|
1|   |   |   | 4 |   |   |        |  |11|12|         |
  |-----|---|---|---|---|--------|--|--|--|---------|
2|     SYSDB - RELATIVE POINTER TO THE DIT         |
  |-----|--|--------------------------------------|
3|     |AR|                                         |
  |-----|--|--------------------------------------|
```

4 - 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).

11 - 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.

12 - Word 1.(12:1) - Part of Device Subtype field. If true, device may be
allocated automatically when opened. If false, operator

AR - Word 3.(2:1) - must allocate. AUTO REPLY. Device may be allocated
without prompting the operator for REPLY if certain
run-time conditions are met; this bit is set
automatically if word 1.(12:1) is TRUE.

Entry for Disc Drives

```
   0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
  |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
0| 0|                                              |
  |--|--|--|--|--|--|--|--|--|--|--|--|--|--|------|
1| 0 |   |   | 4 | 5| 6|        |10|11|             |
  |-----|---|---|---|--|--|--------|--|--|----------|
2|     SYSDB - RELATIVE POINTER TO THE DIT         |
  |--|--|--|--------------------------------------|
3|  |SD|AR|                                         |
  |--|--|--|--------------------------------------|
```

0 - 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.

4 - Word 1.(4:1) - If true, the disc is a nonsystem domain (private volume,
serial disc or foreign disc) disc drive.

5 - Word 1.(5:1) - If true, disc is a mounted private volume.

6 - Word 1.(6:1) - If true, the disc is a reserved volume used to satisfy
the requirements of a multiple volume private volume
set.

10 - 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.

11 - Word 1.(11:1) - If bit 10 is true, then 1 ==> foreign disc, 0 ==> serial
disc.

SD - 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.

AR - Word 3.(2:1) - AUTO REPLY (serial or foreign disc only) Device may be
allocated without prompting the operator if certain
run-time conditions are met.

Logical Device Table (LDT)

Overview of Data Segment

```
| DST 14 (X16)  |---------------------|<-----DST X16
| SIR 10 (X12)  |                     |
               |  LOGICAL DEVICE     |
               |      TABLE          |
               |      (LDT)          |
               |---------------------|
               |  LOGICAL DEVICE     |
               |  TABLE EXTENSION    |
               |      (LDTX)         |
               |                     |
               |---------------------|
```

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

Typical Entry Format

```
   0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
  |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
| 0|       FILE USE COUNT                          |
  |-----------------------------------------------|
| 1|    VOLUME TABLE INDEX IF DEVICE TYPE = 0-7,   |
|  |         ELSE MAIN PROCESS PIN #,              |
|  |         OR SPOOLER PROCESS PIN #              |
  |-----------------------------------------------|
| 2|    RECORD WIDTH         |CS|FO|  DEVICE TYPE  |
  |-----|--|--|--|--|--|--|--|--|-----------------|
| 3|SPOOL|SY|DI|DN|TR|HD|CL|SQ| DEVICE - DEPENDENT |
  |--|--|--|--|--|--|--|--|--|--------------------|
| 4|LK|        XDD HEAD INDEX                      |
  |--|--------------------------------------------|
| 5|    CONTROL-Y PIN                              |
  |-----------------------------------------------|
| 6|    DEFAULT OUTPUT DEVICE -OR-                 |
|  |         DEFAULT CLASS INDEX                   |
  |-----------------------------------------------|
```

CS - Word 2.(8:1) - Communication system device if set.
FO - 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.
SY - Word 3.(2:1) - Device is available to system (not down).
DI - Word 3.(3:1) - Device is available to diagnostics (obs).
DN - Word 3.(4:1) - :DOWN requested, honored when use count = 0.
TR - Word 3.(5:1) - If set, trailers are disabled.
HD - 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.
CL - Word 3.(7:1) - If I/O, word 6 is the Device Class Table
index/LDEVN of the default output class/device
associated with this device.
SQ - Word 3.(8:1) - Spooling has been enabled (spool queues are
open) for this device.
Word 3.(9:1) - Device dependent information:
1. For terminal-like devices, the default
terminal type to be used if not specified
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
              unlabeled tapes only.
              For either:
                  0 = unknown density/no FOPEN w/write
                  1 = 1600 BPI
                  2 = 6250 BPI
                  3 = 800 BPI
Word 4.(0:1) - Auxiliary lock mechanism - if set, device is
              being deallocated but LDT and XDD are
              inconsistent and their SIRs are released;
              :ALLOCATE cooperates by rejecting access to
              the device.

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                       |
   |--|--|--|--|--|--|-----------------------------|
 1 |        DEVICE-SPECIFIC INFORMATION FIELDS      |
   |-                                              -|
 2 |-                                              -|
   |-                                              -|
 3 |      REFER TO THE FOLLOWING EXAMPLES           |
   |-                                              -|
 4 |         (LDTX ENTRIES)                         |
   |-----------------------------------------------|
```

Where:

 S = Seek ahead enable/disable flag (system or PV disc only).
 SD = This logical device is a Serial or 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 |        CHANNEL ID                              |
   |-----------------------------------------------|
 3 |        NLIO XDS                                |
   |-----------------------------------------------|
 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 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.

NLIO XDS = Extra Data Segment Number of Working storage for
           NLIO translation.

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: XDSM 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 deactivated 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 (X50)  |-----------------------|  DST X50
| SIR 40 (X50)  |                       |
               |   DEVICE CLASS TABLE   |
               |                       |
               |         (DCT)         |
               |                       |
               |-----------------------|
               |                       |
               |  TERMINAL DESCRIPTOR  |
               |         TABLE         |
               |         (TDT)         |
               |-----------------------|
```

Header Entry Format

```
     0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
   |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
 0|           DCTH'SEGMENT'SIZE                     |
   |------------------------------------------------|
 1|           DCTH'ENTRY'SIZE                       |
   |------------------------------------------------|
 2|           DCTH'NUM'DCT'ENTRIES                  |
   |------------------------------------------------|
 3|           DCTH'DCT'BASE (SET TO 6)              |
   |------------------------------------------------|
 4|           DCTH'NUM'TDT'ENTRIES                  |
   |------------------------------------------------|
 5|           DCTH'TDT'BASE                         |
   |------------------------------------------------|
```

Device Class Table 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  |SO| T|CLASS ACCESS TYPE|
    |--|---------------------|--|--|----------------|
  5|          DCT'NUM'DEVICES                        |
    |------------------------------------------------|
  6|          LDEV #1                                |
    |------------------------------------------------|
  7|          LDEV #2                                |
    |------------------------------------------------|
                    .                    .
                    .                    .
                    .                    .
    |------------------------------------------------|
|n+5|          LDEV # n                              |
    |------------------------------------------------|
```

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.

Some fields in the DCT require further description, as follows:

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 request 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 by the class (0-7 for disc, 24 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 type (31, or X37), 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 hardware type is kept in the LDT, as for serial discs.

Terminal Descriptor Table Typical Entry Format

```
        0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
        |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
  |   0|                                              |0
  |   1|                                              |1
  |    |      DESCRIPTOR FILE NAME (ASCII)            |
  |   2|                                              |2
  |    |                                              |
  |   3|                                              |3
  |    |----------------------------------------------|4
  |   4|                                              |
  |    |                                              |5
  |   5|          GROUP NAME                          |
  |   6|                                              |6
  |    |                                              |
  |   7|                                              |7
  |    |----------------------------------------------|8
  |  10|                                              |
  |    |                                              |9
  |  11|          ACCOUNT NAME                        |
  |  12|                                              |10
  |    |                                              |
  |  13|                                              |11
  |    |----------------------------------------------|
  |  14|   NUMBER OF DEVICES IN USING FILE (n)        |12
  |    |----------------------------------------------|
  |  15|   LDEV #1                                    |13
  |    |----------------------------------------------|
  |  16|   LDEV #2                                    |14
  |    |----------------------------------------------|
            .                    .
            .                    .
            .                    .
  |        |----------------------------------------------|
  |n+X14|   LDEV #n                                    |n+12
         |----------------------------------------------|
```

The Terminal Descriptor Table contains a varying number of variable length
entries, because each Terminal Descriptor entry may have an arbitrary number
of logical devices. However, you can only configure a fixed number of valid
terminal entry files. These are the TTnn or TTPCLnn files which reside in
PUB.SYS. SYS is one of these files.

Interrupt Linkage Table (ILT) for HP-IB Systems

```
        0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
        |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
  X0|      CHANNEL PROGRAM VARIABLE AREA              |ICPVA0    (0 for ATP)
   1|      (ICPVA) FOR TERMINALS WITH ATP             |ICPVA1    (0 for ATP)
   2|      DRIVERS, THIS AREA IS ZERO                 |ICPVA2    (0 for ATP)
   3|                                                 |ICPVA3    (0 for ATP)
     |-------------------------------------------------|
   4|      DMA ABORT ADDRESS                          |ICPVA4
  | |---                                          --- |
   5|                                                 |ICPVA5
     |-------------------------------------------------|
   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|UP|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 SIARTIO 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 DITs 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|RC|RQ|SI|MU| O|IO|IA|NO|ST|NS|   STATE     |DFLAG
     |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
  1|    SYSDB RELATIVE POINTER TO THE DIT FOR         |DLINK
     |      THE NEXT DEVICE REQUESTING                |
     |      THIS RESOURCE OR SERVICE                  |
     |-------------------------------------------------|
  2|    SYSDB RELATIVE POINTER TO THE FIRST           |DIOQP
     |      IOQ IN REQUEST LIST                       |
     |      FOR THIS DEVICE                           |
     |-------------------------------------------------|
  3|    LOGICAL DEVICE NUMBER                         |DLDEV
     |-------------------------------------------------|
  4|    SYSDB RELATIVE POINTER TO DEVICE              |DDLTP
     |      LINKAGE TABLE                             |
     |-------------------------------------------------|
  5|    SYSDB RELATIVE POINTER TO INTERRUPT           |DILTP
     |      LINKAGE TABLE                             |
     |-------------------------------------------------|
  6|    CONTROLLER HARDWARE STATUS                    |DSTAT
     |-------------------------------------------------|
  7|    HARDWARE ERROR STATUS.  SET WHEN THE          |DSERR
     |      DRIVER DETECTS AN ERROR.  WHENEVER        |
     |      <>0, THE DRIVER MONITOR LOGS AN           |
     |      AN I/O ERROR AND CLEARS THIS WORD         |
     |-------------------------------------------------|
 10|    DEVICE DEPENDENT AREA                         |DTIME
     |-------------------------------------------------|
 11|    DEVICE DEPENDENT AREA                         |DTRQX
     |-----|-------------------------------------------|
 12| IOT |              | PHYSICAL 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 the
        monitor is active.
    MU - If set, indicates device is a multiple unit
        controller.
    IO - If set, a channel program is currently executing.
    IR - If set, an interrupt or response has occurred.
    NO - If set, device is in a not ready or operator wait
        state.
    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 (reserved)
    2 - Call driver initiator
    3 - Call driver completor
    4 - Not used (reserved)
    5 - Complete request
    6 - Unexpected interrupt occurred
    7 - Start operator intervention wait
    10 - Operator wait; restart at 0
    11 - Data makepresent/freeze wait
    12 - Initiator code makepresent/freeze wait
    13 - Interrupt completion wait
    14 - Device controller availability wait
    15 - Not used (reserved)
    16 - Initiator code makepresent wait
    17 - Completor code makepresent wait
DUNIT - I/O system type and unit number.
    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|IR|NO|ST| 0|   STATE    |DFLAG
 1|        SYSDB RELATIVE POINTER TO THE DIT          |DLINK
  |        FOR THE NEXT DEVICE REQUESTING             |
  |        THIS RESOURCE OR SERVICE                   |
  |--------------------------------------------------|
 2|        IOQ TABLE INDEX TO THE FIRST IOQ           |DIOQP
  |        IN THE REQUEST LIST FOR                    |
  |        THIS DEVICE                                |
  |-------------------------|------------------------|
 3| PHYSICAL UNIT #         |  LOGICAL DEVICE NUMBER  |DLDEV
  |-------------------------|------------------------|
 4|        SYSDB RELATIVE POINTER TO                  |DDLTP
  |        DEVICE LINKAGE TABLE                       |
  |--------------------------------------------------|
 5|        SYSDB RELATIVE POINTER TO                  |DILTP
  |        INTERRUPT LINKAGE TABLE                    |
  |--|--|--|--|--|--|--------------------------------|
 6|VS|AB|RE|TP|NR| NR CNT  |    DEVICE STATUS         |DSAVE
  |--|--|--|--|--|----------------------------------|
 7|        HARDWARE ERROR STATUS - SET WHEN THE       |DSERR
  |        DRIVER DETECTS AN ERROR;                   |
  |        IF <> 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     |DRQST
  |        WHILE A TIMER IS ACTIVE                    |
  |--|-----------------------|----------------------|
12| IOT |                     |PHYSICAL UNIT NUMBER |DUNIT
  |--|--|--|--|---------|---------|--------|--------|
13|RF|UE|DE|TO|UNIT CNT |DATA CNT | TO CNT |PRTY CNT|DCOUNTS
  |--|--|--|--|---------|---------|--------|--------|
14|        ERROR LOGGING LOCATION #1                  |DLOGERROR
  |--------------------------------------------------|
15|        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.
  IR IRK   - 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 - Logical device number.
DUNIT - I/O system type and unit number.
    0 - HP 3000 Series II/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 UNIT	SIO PREMP	IO PROG	IAK	M HEAD	NT RY			STATE			DFLAG
1	NEXT DITP																DLINK
2	IOQP																DIOQP
3	LOGICAL DEVICE NUMBER																DLDEV
4	DLTP																DDLTP
5	ILTP																DILTP
6	CONTROLLER HARDWARE STATUS																DSTAT
7	HARDWARE ERROR STATUS																DSERR
10																	DTIME
11																	DTRQX
12	IOT		2					7			PHYSICAL 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 RY   - Not ready for SIO; SIODM holds off next SIO until
               ALLOWPOLL is done.

DIT for Channel Devices (Cont.)

DFLAG.STATE - This quantity specifies the next action to be taken
              in servicing the request.

                X0 - 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 (X10)
                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   - Logical Device Number.
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   - Timeout completed flags. If a timeout occurs in response
           to a timer request type X20 (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 W8 if timer
           services are requested.)
DTRQX   - Used by some device drivers, it denotes timer request index.
DUNIT   - I/O system type and unit number.
  .UNIT - Unit number of the physical device.
  .IOT   - IO type 0= Series III I/O; 1= HPIB I/O

DIT For 7905/7906/7920/7925

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15		
0	0	1	ACT	REQ	CD	M UNIT	O	I/O PROG	IAK	1	0	0		STATE			0	DFLAG
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
10	INDEX OF FIRST REQUEST IN QUEUE																8	DMARQ
11	INDEX OF LAST REQUEST IN QUEUE																9	DMARQT
12	IOT							PHYSICAL UNIT #									10	DUNIT
13	SIO PROGRAM-RELATIVE ABORT ADDRESS																11	DLOGSIOP
14	CURRENT PHYSICAL																12	CPDA
15	DISK ADDRESS																13	
16	CURRENT DATA BUFFER ADDRESS																14	CDBA
17	WORD COUNT REMAINING																15	WCR
20	CURRENT WORD COUNT																16	CWC
21	SYSBUF INDEX																17	SYSBUFA
22	STATUS 1 RETURN																18	STAT1
23	STATUS 2 RETURN																19	STAT2
24	CYL																20	CEDA
25	HEAD								SECTOR								21	
26	STATUS 1 RETURN																22	
27	CYL																23	

DIT For 7905/7906/7920/7925 (Cont.)

30	HEAD	SECTOR	24	> REQUEST
31	DISPLACEMENT		25	SYNDROME
32	PATT 1		26	
33	PATT 2		27	
34	PATT 3		28	
35	SECTOR COUNT TO TRANSFER		29	SCOUNT
36	INITIALIZE ADDRESS		30	INITADR
37			31	
40			L 32	DMISC
41	CONTROLLER STATUS AFTER SEEK		33	SEEKSTAT
42	IN CHANNEL PROGRAM		34	
43	CPVA WORD 0 UPON CHANNEL ABORT		35	DLOGERROR
44	CURRENT LOGICAL SECTOR ADDRESS		36	CLDA
45	CURRENT LOGICAL SECTOR ADDRESS		37	

DMISC
  (15:1) L'STAT'ERR - 1 Last transfer ended in error.

DUNIT - I/O system type and unit number.
  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| SI| E| M| W| T| O| C|CL| O| O| O| O| RETRY  |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
   |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
 XO|TM|DS|AC|RQ|CD| O| O|IO|IA|NO|ST| O|   STATE   |DFLAG
   |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
 1 |    SYSDB RELATIVE POINTER TO THE DIT          |DLINK
   |    FOR THE NEXT DEVICE REQUESTING             |
   |    THIS RESOURCE OR SERVICE                    |
   |-----------------------------------------------|
 2 |         CURRENT REQUEST INDEX                 |DCURREQP
   |-----------------------------------------------|
 3 |         LOGICAL DEVICE NUMBER                 |DLDEV
   |-----------------------------------------------|
 4 |    SYSDB RELATIVE POINTER TO DEVICE           |DDLTP
   |         LINKAGE TABLE                          |
   |-----------------------------------------------|
 5 |    SYSDB RELATIVE POINTER TO INTERRUPT        |DILTP
   |         LINKAGE TABLE                          |
   |-----------------------------------------------|
```

CS 80 Disc Device Information Table (DIT) (Cont.)

```
   |-----------------------------------------------|
 6 | DSTAT IS -1 WHEN A SYSTEM POWERFAIL           |DSTAT
   |    OCCURRED                                    |
   |-----------------------------------------------|
 7 | HARDWARE ERROR STATUS.  SET WHEN THE          |DSERR
   |    DRIVER 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 |
   |-----------------------------------------------|
14 | HIGH ORDER LOGICAL SECTOR ADDRESS OF BAD BLOCK|DBADBLK1
   |-----------------------------------------------|
15 | LOW ORDER LOGICAL SECTOR ADDRESS OF BAD BLOCK |DBADBLK2
   |-----------------------------------------------|
16 | BYTE TRANSFER LEFT WHEN BAD BLOCK OCCURRED    |DBADXFER
   |-----------------------------------------------|
17 | HARDWARE LOGGED ERROR STATUS - CPVR (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     |RSPWORD2
   |----------------------------------|------------|
```

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 IRK     - An interrupt or response has occurred for this device.
NO NOTRDY  - Go to the state X10 after Idle Channel Program is stated.
```

```
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:

            X0 - 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 and unit number.

```
            0 - Non-HP-IB
            1 - HP 3000 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.
```

DSBUFRDDR - 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 - CPVR(0) logged on hardware error status.

DSIOPSTOP - Stopped channel program relative offset location due
            to an error in CPVR(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; does not return hard error.
Local State - State of the local request made by driver:

           0 - No local request is being processed.
           1 - Reading revision 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	0	ACT	REQ	0	M UNIT	0	0	I/O PROG	IAK	0	0	0	STATE		DFLAG

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

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
10	TIMEOUT FLAGS	DTIME
11	TIMER REQUEST INDEX	DTRQX
12	IOT .... PHYSICAL UNIT #	DUNIT
13	.... R4 RW	DDFLAGS

DUNIT - I/O system type and unit number.
    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.
      DDFLAGS - Device dependent flags.
  R4 DDFLAGS - (bit 14) if set, need to rewind tape before next write.
  RW DDFLAGS - (bit 15) if set, tape is rewound.

QMISC

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R	B	F	G	E	S	U	FORWARD SPACE COUNTER			BACK SPACE COUNTER			RETRY COUNTER		

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 7974/78 Magnetic Tape Drives

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.

(bits 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 device requesting this resource or service	DLINK
2	SYSDB relative pointer to the first IOQ in request list for this device	DIOQP
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 PF EOV PA	DSAVE
7	Hardware error status. Set when the driver detects an error. Whenever <>0, the driver monitor logs an I/O error and clears this word	DSERR
X10	Bit 0 is set at completion of timer	DTIME
X11	Interrupt status for this unit. Set by the driver each time it processes an interrupt.	DSTAT
X12	IOT /////////// Physical unit number	DUNIT
X13	Holds the time out request entry index while a timer is active.	DRQST
X14	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 X10 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
                 X10 - wait for interrupt (operator intervention)
                     restart at state 0
                 X11 - wait for data segment freeze, then state 2
                 X12 - wait for driver initiator to be frozen, then
                     allocate controller (state 2)
                 X13 - wait for I/O completion interrupt, then state 3
                 X14 - wait for controller, then call driver initiator
                 X15 - not used
                 X16 - wait for initiator wake present, then state 2
                 X17 - wait for completor wake 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 transfered from the short-read buffer to the
              user's buffer.
  PF POWER - Device power up indication.
  PA PENDING ABORT - An abort is pending for a command queued IOQ.
  FO FIRST OPERATION - The first read or write after a rewind
              command is not done in queuing mode.
  EOV End'Of'Volume - enable check on 2 consecutive EOFs.

|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 *

DIT for 7979/80 Magnetic & DAT Tape Drives

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 array.

Z	0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15	MNEMONIC
0	O| O|AC|RQ| O|MU| O|IO|IR|NO|ST| O| State	DFLAG
1	SYSDB relative pointer to the DIT for the next device requesting this resource or service	DLINK
2	SYSDB relative pointer to the first IOQ in request list for this device.	DIOQP
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|RO|SH|GR| |PF| |EOV|PP|IR| EI |PR|	DSAVE
7	Hardware error status. Set when the driver detects an error. Whenever <> 0, the driver monitor logs an I/O error and clears this word	DSERR
10	Bit 0 is set at completion of timer.	DTIME
11	Interrupt status for this unit. Set by the driver each time it processes an interrupt.	DSTAT
12	IOT |////////////////////| Phys. unit #	DUNIT
13	Holds the time out request entry index while a timer is active.	DRQST
14 15 16	Error log. Contains 6 bytes of status from the previous operation.	DDEVSTAT

DFLAG - Device Flags and Request State.
    AC - A monitor is currently servicing this device.
    RQ - A service request is pending while the monitor is active.
    MU - This device is on a multi-unit controller.
    IO - An I/O Channel Program is running for this device.
    IR - An interrupt or response has occurred for this device.
    NO - Not ready, start Idle Channel Program then go to state Z10.
    ST - The device monitor is starting an idle channel program for this device. There is no IOQ associated with this state.

STATE - Device Monitor State.
    Specifies the next action to be taken by SIGDM in servicing the request:
        0 - Start a new request.
        1 - Not used.
        2 - Call driver initiator procedure.
        3 - Call driver completor procedure.
        4 - Not used.
        5 - Completed request processing.
        6 - Initiate device recognition sequence.
        7 - Start operator intervention wait.
        Z10 - Wait for interrupt (operator intervention), restart at state 0.
        Z11 - Wait for data segment freeze, then state 2.
        Z12 - Wait for driver initiator to be frozen, then allocate controller (state 2).
        Z13 - Wait for I/O completion interrupt, then state 3.
        Z14 - Wait for controller, then call driver initiator.
        Z15 - Not used.
        Z16 - Wait for initiator make present, then state 2.
        Z17 - Wait for completor make present, then state 3.

DSAVE - Device processing flags.
    RW - Indicates tape has been rewound.
    RO - Indicates a rewind/offline was performed to allow a write-ring mount.
    SH - Indicates a short read is in progress. After completion of the read, EOF is checked for and if not present, the data requested is transferred from the short read buffer to the user's buffer.
    GR - Good retries on previous operation.
    PF - Indicates device is powered up.
    EOV - enable check on 2 consecutive EOFs.
        0 - no check
        1 - enable
        2 - 1 EOF read
        3 - 2 consecutive EOFs encountered
    PP - Device powerfail processing flag.
        0 - Device powerfail process complete.
        1 - 1st pass of device powerfail processing.
    IR - Immediate report status.
        0 - Immediate report is disabled.
        1 - Immediate report is enabled.
    EI - EOF processing indicator.
        0 - Last operation was not a write file mark.
        1 - Last operation was a write file mark.
        2 - Device buffered operations are being completed prior to issuing the second of a double write file mark (EOF).
    PR - Pending abort processing.

DSTAT - First two bytes of device status.

WORD 1

Bit   Meaning

0   End-of-file (EOF)
1   Beginning-of-tape (BOT)/Load-point (LP)
2   End-of-tape (EOT)
3   Recovered error (STE)
4   Command reject
5   File protect (not write enabled; no write ring)
6   Unrecovered error (MTE)
7   Unit online
8   GCR (6250 BPI Density)
9   Unknown density
10   Data parity error
11   Timing error
12   Tape runaway
13   Door open
14   Not used
15   Immediate report enable

WORD 2

Bit   Meaning

0   PE (1600 BPI Density)
1   NRZI (800 BPI Density)
2   Power restored
3   HPIB Command Parity Error
4   Position Unrecovered
5   Formatter error
6   Servo error
7   Controller error
8-10   Command Reject detail
        000 - Null code
        001 - Reserved
        010 - Device Reject
        011 - Protocol Reject

        100 - Reserved
        101 - Prior error reject
        110 - Reserved
        111 - Selftest failure
11-15   Retry count.

WORD 3 (left byte)

The contents of this byte contains binary coded information
regarding the specific error encountered.

If COMMAND REJECT

5 - Device is write protected when a write type command was
    initiated.
6 - Tape was not tensioned when the command was queued.
7 - Write density command given but requested density is not
    available (option not present).
9 - The tape to be read was unidentifiable as to format.
    The density read may not be available, or the tape may
    have an unreadable identifications field, or may be
    blank
10 - The tape to be written on has not been identified as to
    format. A write Record, Write File Mark, or Write Gap
    command was received but cannot be processed without a
    Write Format command if the tape was unidentified at
    load point.
11 - Drive not online.
16 - A write format command was issued but the tape isn't at
    load point.
19 - A backward type command (except a rewind command) was
    just initiated but the tape was already positioned at
    BOT.
22 - An improper command sequence was detected by the drive.
23 - Protocol not synced.
24 - The tape command byte received was unknown to the drive.
31 - The length of a write record requested exceeded the
    size of the drive's data buffer.
37 - Cannot write past 10 feet beyond end-of-tape.
40 - Door open reject. The door was opened during a long gap
    while the tape was beyond the end-of-tape marker. This
    condition is non-retriable to prevent unspooling of the
    tape.

If UNRECOVERED ERROR

41 - Tape velocity was out of specification.
43 - Tape tension was out of specification.
45 - Multiple tracks were in error. Either two or more
    tracks were in error for a PE or NRZI write, or two
    or more tracks were in error for a GCR write.
47 - Failure to verify a tape mark or density ID just written.
48 - Noise on detect. Indistinguishable flux transitions were
    detected while attempting to detect a recorded block.
49 - Data format error. Flux transitions were found or were
    missing in the appropriate tracks for a block detect.
50 - Failure to identify tape following a rewind command
51 - Gap detected before end-of-data. The read formatter
    dected a full tape width dropout within the data portion
    of a data block.
52 - Data block dropout. A full tape width dropout was
    detected within the preamble or postamble of a data
    block.
53 - Redundancy check error. The read formatter detected
    either a CRC, ACRC, LRC or residual error while reading
    or verifying a data block.
54 - Read parity error. The read formatter detected an
    unrecovered parity error within a data block. For PE
    this error could include multiple tracks in error, and
    for GCR this error could also include a redundancy check
    error. (Buckhorn only).
55 - Abnormal command abort, door opened (Antelope only).
57 - Maximum skew exceeded (Antelope only).
58 - False preamble or postamble detected (Antelope only).
59 - Corrected data error on write (Antelope only).
61 - Data block timeout. Could not detect the gap following
    a data block. Could be caused by a record length longer
    than the drive supportes on read.
62 - Tape mark dropout. A full tape width dropout was
    detected within a tape mark.
63 - Tape mark unverified. A tape mark was detected which
    does not meet ANSI specifications in terms of flux
    transitions and erasure in the appropriate tracks.
64 - Tape mark timeout. Could not detect the gap following
    a detected tape mark.

If POSITION UNRECOVERED
81 - Servo controller unresponsive. The servo will not take
    data from the master controller.
82 - Servo failed to reach the desired state reqquested by
    the master controller.
83 - Servo shutdown. The servo system lost tape tension
    unexpectedly.
84 - Servo controller hard failure. The servo controller has
    detected a hard failure within itself.
85 - Servo protocol error. An invalid byte was received by
    the servo from the master controller.
86 - A run time error was detected by the servo.
87 - In position interrupt not received. Master controller
    did not get the in position interrupt it expected.
88 - No gap detected by the servo after reading or writing a
    data block or tape mark.
90 - No BOT detected on load or rewind.
91 - Speed out of specifications.
92 - The desired state requested by the master controller
    was invalid for the current context.
94 - Tape positioning failure.
If FORMATTER ERROR
101 - Buckhorn read formatter unresponsive. The read formatter
    did not respond with end of record status after a data
    block was detected.
102 - Buckhorn read formatter hardware error.
103 - Bad block type detected on a write operation.
104 - Erase failure. Flux transitions were detected in a
    portion of tape currently being erased.
105 - No data detected after write.
106 - Tracks out of sync on write verify.
107 - Antelope formatter hardware error.
108 - Antelope formatter unresponsive.
110 - Formatter byte count mismatch with data buffer.
If CONTROLLER ERROR
121 - Transaction ID mismatch between command sent to Device
    program and the returned report.
122 - No pending command found for report received from Device
    program.
123 - Invalid report message received from Device program.
124 - Report queue overflow.
125 - Unknown command received by Device program.
126 - Command queue overflow.
128 - Missing end-of-record flag in data buffer.
131 - Byte count mismatch between putting a record into the
    data buffer and removing it.
133 - Processor handshake abort between HP-IB interface board
    and channel program.
134 - Unknown HP-IB interface exception detected.
138 - Device program firmware error.
139 - Hardware utilities firmware error.
140 - Channel program firmware error.

If COMMAND REJECT and PROTOCOL REJECT

161 - Command queue not empty. Cannot accept new tape
      command or diagnostic request.
162 - Request OSJ expected.
163 - Request status expected.
165 - Unknown unit select.
166 - Tape command secondary expected.
167 - Data byte expected.
168 - Missing EOI on tape command data byte, selftest number,
      or END command data byte.
170 - Command phase protocol error for write record.
172 - Read record report phase protocol error.
173 - Report phase protocol error.
174 - Cold load sequence protocol error.
176 - END "Complete" or "Complete-Idle" exptected.
178 - END "Data" expected.
180 - Unknown interface secondary command.
181 - Misplaced data byte.
184 - Interface loopback protocol error.
185 - Run selftest protocol error.
188 - HP-IB command parity error.
189 - Reset by operator during a protocol sequence.
190 - Device clear received. (Internal error code only.)

WORD 3 (second byte)

The sixth byte is used only when reporting transparent status of
hard and soft errors while in immediate report mode. When an
immediate report write has a soft error (retries were necessary)
or a hard error (write failure) this byte indicates which
command had the error. It contains the number of commands sent
and reported since the command in question was issued. If the
immediate reported write had a hard error, all of the commands
issued after the failure also fail (they will be aborted by the
device). Thus on a hard error, this byte actually indicates the
number of preceeding commands that failed. For non-transparent
status, this byte will always be zero.

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
    |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
 X0| 0| 0|AC|RQ| 0|MU| 0|IO|IA| 0| 0| 0|    STATE    |DFLAG
    |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
  1|      SYSDB RELATIVE POINTER TO THE DIT FOR       |DLINK
   |        THE NEXT DEVICE REQUESTING THIS           |
   |        RESOURCE OR SERVICE                        |
   |--------------------------------------------------|
  2|      SYSDB RELATIVE POINTER TO THE FIRST         |DIOQP
   |        IOQ IN THE REQUEST LIST FOR               |
   |        THIS DEVICE                                |
   |--------------------------------------------------|
  3|      LOGICAL DEVICE NUMBER                       |DLDEV
   |--------------------------------------------------|
  4|      SYSDB RELATIVE POINTER TO THE DEVICE        |DDLTP
   |        LINKAGE TABLE                              |
   |--------------------------------------------------|
  5|      SYSDB RELATIVE POINTER TO THE               |DILTP
   |        INTERRUPT LINKAGE TABLE                    |
   |--|--|--|--|--|--------------------------------|
  6|RU|RU|SH| |DC|PF|                                 |DSAVE
   |--|--|--|--|--|--|----------------------------|
  7|      HARDWARE ERROR STATUS                       |DSERR
   |        SET WHEN THE DRIVER 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             |DSTAT
   |        SET BY THE DRIVER EACH TIME IT           |
   |        PROCESSES AN INTERRUPT                    |
   |------|----------------------|------------------|
 12| IOT  |                      | PHYSICAL UNIT #  |
   |------|----------------------|------------------|
 13|      HOLDS THE TIME OUT REQUEST ENTRY INDEX     |DRQST
   |        WHILE A TIMER IS ACTIVE                   |
   |--------------------------------------------------|
 14|      ERROR LOG - CONTAINS 5 VALID BYTES         |DLOGERROR
   |        OF STATUS                                 |
   |--------------------------------------------------|
```

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 X10 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:
                  X0 - 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.
  RU RUBIT   - 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 *

DIT for 9144 Cartridge tape drive

word	0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15	MNEMONIC
0	|TM|DS|AC|RQ| 0| 0| 0|IO|IA|NO|ST| 0| State	DFLAG
1	SYSDB relative pointer to the DIT for the next device requesting this resource or service	DLINK
2	Pointer to the current IOQ	DIOQP
3	Logical device number	DLDEV
4	SYSDB relative pointer to Device Linkage Table	DDLTP
5	SYSDB relative pntr to Interrupt Linkage Table	DILTP
6	Set to -1 when system powerfail occurs.	DSTAT
7	Hardware error status. Set when the driver detects an error. Whenever <> 0, the driver monitor logs an I/O error and clears this word	DSERR
10	index of first request in the queue	DQHEAD
11	index of last request in the queue	DQTAIL
12	IOT		Physical Unit #	DUNIT
13	|LK|IG|IM|	SUBSTATE	DMISC
14	High order logical sector address of bad block	DBADBLK1
15	Low order logical sector address of bad block.	DBADBLK2
16	Byte transfer left when bad block occured	DBADXFER
17	Hardware logged error status - CPVA (0).	DLOGERROR
20	Relative offset of channel program abort.	CSIOPSTOP
21	Accum byte count of transfer > 6144 bytes.	DBYTECNT
22		
23	Device status (20 bytes), errors logged	DSTATUS
.		
.		
34	.	

DFLAG - Device flags and request state.
- TM - Set if device is a terminal
- DS - If TM = 0 and this bit is set then device is a disc, otherwise device dependent.
- AC - A monitor is currently servicing this device.
- RQ - A service request is pending while the monitor is active.
- IO - An I/O channel program is running for this device.
- IA - An interrupt or response has occurred for this device.
- NO - Not ready, start idle channel program then go to state X10.
- ST - The device monitor is starting an idle channel program for this device. There is no IOQ associated with this state.
- STATE - State of the device monitor. Specifies the next action to be taken by SIODM in servicing the request:
  - 0 - Start a new request.
  - 1 - Not used.
  - 2 - Call driver initiator procedure.
  - 3 - Call driver completor procedure.
  - 4 - Not used.
  - 5 - Request complete.
  - 6 - Initiate device recognition sequence.
  - 7 - Start operator intervention wait.
  - X10 - Wait for interrupt (operator intervention), restart at state 0.
  - X11 - Wait for data segment freeze, then state 2.
  - X12 - Wait for driver initiator to be frozen, then allocate controller (state 2).
  - X13 - Wait for I/O completion interrupt, then state 3.
  - X14 - Wait for controller, then call driver initiator.
  - X15 - Not used.
  - X16 - Wait for initiator make present, then state 2.
  - X17 - Wait for completor make present, then state 3.

DLINK,
DQHEAD,
DQTAIL,
DUNIT - Not used.

DMISC - Miscellaneous device information.
- LK - Lock flag denoting unload status of the device.
  - 0 - Allow operator unload of the volume.
  - 1 - Deny operator unload of the volume.
- IG - Ignore unexpected interrupt flag.
- IM - Immediate report.
  - 0 - Disabled.
  - 1 - Enabled.
- SUBSTATE - Idle channel program state.
  - 0 - Normal idle channel program wait.
  - 1 - Idle request being serviced wait.

DBADBLK1 - High order logical sector address of bad block encountered.

DBADBLK2 - Low order logical sector address of bad block encountered.

DBADXFER - Byte transfer left when bad block occured.

DLOGERROR - CPVA (0) logged on hardware error status.

DSIOPSTOP - Relative offset location of channel program when error in CPVA (0) occured.

DBYTECNT - Accumulative transfer count for transfers greater than 6144 bytes.

DSTATUS - 20 bytes of status logged when a status error occurs. (Refer to CS/80 Instruction Set manual for description.)

DIT for HP9145 Cartridge Tape Drive

word	0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15	MNEMONIC
0	|TM|DS|AC|RQ| 0| 0| 0|IO|IA|NO|ST| 0| State	DFLAG
1	SYSDB relative pointer to the DIT for the next device requesting this resource or service	DLINK
2	Pointer to the current IOQ	DIOQP
3	Logical device number	DLDEV
4	SYSDB relative pointer to Device Linkage Table	DDLTP
5	SYSDB relative pntr to Interrupt Linkage Table	DILTP
6	Set to -1 when system powerfail occurs.	DSTAT
7	Hardware error status. Set when the driver detects an error. Whenever <> 0, the driver monitor logs an I/O error and clears this word	DSERR
10	index of first request in the queue	DQHEAD
11	index of last request in the queue	DQTAIL
12	IOT	|RV|	Physical Unit #	DUNIT
13	|LK|NR|IM|PI| |SL|MP|	SUBSTATE	DMISC
14	High order logical sector address of bad block	DBADBLK1
15	Low order logical sector address of bad block.	DBADBLK2
16	Byte transfer left when bad block occurred	DBADXFER
17	Hardware logged error status - CPVA (0).	DLOGERROR
20	Relative offset of channel program abort.	CSIOPSTOP
21	Accum byte count of transfer > 6144 bytes.	DBYTECNT
22		
23	Device status (20 bytes), errors logged	DSTATUS
.	used by Request Status Function	
.		

```
     |--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--|
 35  | Device Powerfail Status (4 words)            |    DPFSTATUS
     |--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--|
  .  | used by the Return Powerfail Status Utility  |
  .  |--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--|
  .  |                                              |
  .  |--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--|
 40  |                                              |
     |--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--|
```

DFLAG - Device flags and request state.
- TM   - Set if device is a terminal
- DS   - If TM = 0 and this bit is set then device is a disc, otherwise device dependent.
- AC   - A monitor is currently servicing this device.
- RQ   - A service request is pending while the monitor is active.
- IO   - An I/O channel program is running for this device.
- IA   - An interrupt or response has occurred for this device.
- NO   - Not ready, start idle channel program then go to state X10.
- ST   - The device monitor is starting an idle channel program for this device. There is no IOQ associated with this state.
- STATE - State of the device monitor. Specifies the next action to be taken by SIODM in servicing the request:
  - 0   - Start a new request.
  - 1   - Not used.
  - 2   - Call driver initiator procedure.
  - 3   - Call driver continuator procedure.
  - 4   - Not used.
  - 5   - Request complete.
  - 6   - Initiate device recognition sequence.
  - 7   - Start operator intervention wait.
  - X10 - Wait for interrupt (operator intervention), restart at state 0.
  - X11 - Wait for data segment freeze, then state 2.
  - X12 - Wait for driver initiator to be frozen, then allocate controller (state 2).
  - X13 - Wait for I/O completion interrupt, then state 3.
  - X14 - Wait for controller, then call driver initiator.
  - X15 - Not used.
  - X16 - Wait for initiator make present, then state 2.
  - X17 - Wait for continuator make present, then state 3.

- DLINK,
- DQHEAD,
- DQTAIL - Not used.
- DUNIT - IOT - I/O Type. 01 for this device.
  - RV   - RVR flag. Set by driver, tested by SIODM.
    - 1 - Driver can detect unexpected (RVR) conditions and will exit to SIODM State 6 when safe. SIODM should bypass State 6 when called from GIP. New functionality.
    - 0 - Driver cannot detect RVR condition. SIODM should honor a State 6 call from GIP. Previous functionality.
  - Unit - Unit number for multi-unit controllers. Always 0 for this driver.

- DMISC - Miscellaneous device information.
  - LK   - Lock flag denoting unload status of the device.
    - 0 - Allow operator unload of the volume.
    - 1 - Deny operator unload of the volume.
  - NR   - Not Ready. Remembers the state of DIT'DEV'NOT'RDY the last time status was read from the device. Used for detecting off-line transitions.
  - IM   - Immediate report.
    - 0 - Disabled.
    - 1 - Enabled.
  - PI   - Pending Interrupt. Set if a tape comes online while the driver is processing an IOQ in the continuator. This is checked before the driver enters IDLE in the initiator. If set, the driver will return unexpected interrupt up to SIODM. This will RVR the tape.
  - SL   - Spares Lost. Set if no IOQs are being processed and a pfail has occurred such that the spares table is lost. If this is set when the next IOQ is processed, that IOQ'STAT will be set to X274.
  - MP   - Must Powerfail Next I/O. Set if no I/Os are being processed and a pfail has occurred such that data was lost from the I/O. If this is set when the next IOQ is processed, that IOQ'STAT will be set to X63 POWERFAIL'ABORT.
  - SUBSTATE - Idle channel program state.
    - 0 - Normal idle channel program wait.
    - 1 - Idle request being serviced wait.

DBYTECNT - Accumulative transfer count for transfers greater than 6144 bytes.

DSTATUS - 20 bytes of status logged when a status error occurs. (Refer to CS/80 Instruction Set manual for a more detailed description.)

The following table denotes the only valid status bits that can be set by Excalibur.

WORD 0 : IDENTIFICATION ERRORS FIELD

bits	Meaning	Driver Variable Name
0:4	Volume Number	DIT'FIRST'STAT'WORD
4:4	Unit Number	DIT'FIRST'STAT'WORD
8:8	Status Pending	DIT'UNIT'ATTENTION

WORD 1 : REJECT ERRORS FIELD

bit#	Meaning	Driver Variable Name
2	Channel Parity Error	DIT'CHAN'PARITY'ERR
5	Illegal Opcode	DIT'ILL'OPCODE
6	Module Addressing	DIT'MODULE'ADDR'ERR
7	Address Bounds	DIT'ADDR'BOUNDS
8	Parameter Bounds	DIT'PARAMETER'BOUND
9	Illegal Parameter	DIT'ILL'PARAMETER
10	Message Sequence	DIT'ILL'MSG'SEQ
12	Message Length	DIT'MSG'LENGTH'ERR

WORD 2 : FAULT ERRORS FIELD

bit#	Meaning	Driver Variable Name
6	Unit Fault	DIT'UNIT'FAULT
8	Diagnostic Result	DIT'DIAG'FAIL
14	Powerfail	DIT'DEV'POWERFAIL
15	Retransmit	DIT'RETRANSMIT

WORD 3 : ACCESS ERRORS FIELD

bit#	Meaning	Driver Variable Name
1	Uninitialized Media	DIT'UNINIT'MEDIA
2	No Spares Available	DIT'SPARE'UNAVAIL
3	Not Ready	DIT'DEV'NOT'READY
4	Write Protected	DIT'WRITE'PROTECT
9	Unrecoverable Data	DIT'UNRECOV'DATA
11	End Of File	DIT'END'OF'FILE
12	End Of Volume	DIT'END'OF'VOLUME

WORD 4 : INFORMATION ERRORS FIELD

bit#	Meaning	Driver Variable Name
0	Operator Req Release	DIT'I'OPR'REL'REQ
1	Diagnostic Req Release	DIT'I'DIAG'REL'REQ
7	Auto Sparing Invoked	DIT'DEFECT'BLK'SPARE
11	Recoverable Data	DIT'RECOV'DATA

WORDS 5,6,7,8,9 : PARAMETER FIELD

Refer to the CS'80 Manual for the meaning of these bytes. The bytes depend upon which error is reported in the status bits.

DPFSTATUS - This status is returned by the Return Powerfail Status Utility after a device powerfail. It is valid ONLY directly after the QSTAT=2 report is received from the device. Do not interpret this status if a powerfail has not just occurred.

The Utility returns 7 bytes at present, but 4 words are set aside for it.

BYTE#	Meaning
0	Total Number of bytes returned by Utility, including byte 0 (currently = 7) Driver Variable Name : DIT'PFAIL'NUM'BYTES
1	Powerfail Status Flag Driver Variable Name : DIT'PFAIL'PFSTATUS 0 : If no pfail has occurred (media loaded w/no pfail) 1 : Pfail occurred, but no tape LOADed (tape could be present or UNLOADing but it is not LOADed) 2 : Pfail occurred and a tape has LOADed Successfully 3 : Pfail occurred, tape attempted to LOAD, but LOAD failed
2	Powerfail Data Loss Flag Driver Variable Name : DIT'PFAIL'DATA'LOSS 0 : If no data was lost during last pfail 1 : Host data in buffer was not written to media after pfail (data lost) 2 : Spares table was not updated to tape after pfail (this is FATAL)
3 - 6	Address of First Host Block Not Written Driver Variable Names : DIT'PFAIL'HOST'ADDR1 ... DIT'PFAIL'HOST'ADDR4 If Byte#2 = 1 then this is the logical host block address of the first block not written to tape after the powerfail. If Byte#2 = 0 or 2 then this will be zero.

G.23.00
13- 59

DIT for HP35401 Cartridge Tape Drive

word	0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15	MNEMONIC
0	TM|DS|AC|RQ| 0| 0| 0|IO|IA|NO|ST| 0| State	DFLAG
1	SYSDB relative pointer to the DIT for the next device requesting this resource or service	DLINK
2	Pointer to the current IOQ	DIOQP
3	Logical device number	DLDEV
4	SYSDB relative pointer to Device Linkage Table	DDLTP
5	SYSDB relative pntr to Interrupt Linkage Table	DILTP
6	Set to -1 when system powerfail occurs.	DSTAT
7	Hardware error status. Set when the driver detects an error. Whenever <> 0, during the monitor logs an I/O error and clears this word	DSERR
10	index of first request in the queue	DQHEAD
11	index of last request in the queue	DQTAIL
12	IOT | |AV|RC| RL | | Physical Unit #	DUNIT
13	|LK|NR|IM|PI|RG|LW|MP| | SUBSTATE	DMISC
14	High order logical sector address of bad block	DBADBLK1
15	Low order logical sector address of bad block.	DBADBLK2
16	Byte transfer left when bad block occured	DBADXFER
17	Hardware logged error status - CPVA (0).	DLOGERROR
20	Relative offset of channel program abort.	CSIOPSTOP
21	Accum byte count of transfer > 6144 bytes.	DBYTECNT
22		
23	Device status (20 bytes), errors logged	DSTATUS
·		
·		
34		

G.23.00
13- 60

DFLAG - Device flags and request state.
- TM - Set if device is a terminal
- DS - If TM = 0 and this bit is set then device is a disc, otherwise device dependent.
- AC - A monitor is currently servicing this device.
- RQ - A service request is pending while the monitor is active.
- IO - An I/O channel program is running for this device.
- IA - An interrupt or response has occurred for this device.
- NO - Not ready, start idle channel program then go to state %10.
- ST - The device monitor is starting an idle channel program for this device. There is no IOQ associated with this state.
- STATE - State of the device monitor. Specifies the next action to be taken by SIODM in servicing the request:
  - 0 - Start a new request.
  - 1 - Not used.
  - 2 - Call driver initiator procedure.
  - 3 - Call driver completor procedure.
  - 4 - Not used.
  - 5 - Request complete.
  - 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,
DQHEAD,
DQTAIL - Not used.   <<09421>>

G.23.00
13- 61

DUNIT - IOT - I/O Type. 01 for this device.   <<09421>>
- AV - AVR flag. Set by driver, tested by   <<09421>>
  SIODM.   <<09421>>
  1 - Driver can detect unexpected   <<09421>>
  (AVR) conditions and will exit to   <<09421>>
  SIODM State 6 when safe. SIODM   <<09421>>
  should bypass State 6 when called   <<09421>>
  from GIP. New functionality.   <<09421>>
  0 - Driver cannot detect AVR condi-   <<09421>>
  tion. SIODM should honor a State   <<09421>>
  6 call from GIP. Previous func-   <<09421>>
  tionality.   <<09421>>
- Unit - Unit number for multi-unit control- <<09421>>
  lers. Always 0 for this driver.   <<09421>>
- RC - This bit is the Release count bit.
  It is used to count how many release
  commands have been sent to the Merlin.
  It will either be a 0 or a 1.
  0 - No release command has yet been
  sent to the Merlin.
  1 - A release command has ben sent to
  the Merlin.
  This bit is to insure that we never send
  send more than two releases.
- RL - These are the Release command bits.
  We can send a Release to the Merlin
  from three areas and RL keeps track of
  where Release was sent from.
  0 - No Release command sent.
  1 - Release sent from Initiator se.
  2 - Release sent from Continuator.
  3 - Release sent as a function code.

G.23.00
13- 62

DMISC - Miscellaneous device information.
- LK - Lock flag denoting unload status of the device.
  - 0 - Allow operator unload of the volume.
  - 1 - Deny operator unload of the volume.
- NR - Not Ready. Remembers the state of <<09421>> DIT'DEV'NOT'RDY the last time status <<09421>> was read from the device. Used for <<09421>> detecting off-line transitions. <<09421>>
- IM - Immediate report.
  - 0 - Disabled.
  - 1 - Enabled.
- PI - Pending Interrupt. Device came on <<09421>> line while driver was waiting on com- <<09421>> mand completion (i.e., not in idle <<09421>> CP). On line interrupt was used to <<09421>> complete command, so SIODM didn't <<09421>> wake DEVREC. The interrupt also set <<09421>> this bit. If the bit is set when <<09421>> SIODM starts the idle CP, we return <<09421>> the Unexpected Interrupt state so <<09421>> that SIODM will wake DEVREC. <<09421>>
- RG - Release Granted. Device requested a <<09421>> release and driver O.K.'ed it. If <<09421>> this bit is set when we are ready to <<09421>> start the idle CP, we issue a status <<09421>> request CP instead. This is because <<09421>> the device does not set a DSJ of 2 <<09421>> when completing the release due to <<09421>> coming on line! It needs an addition- <<09421>> al command to do so. The status re- <<09421>> quest CP ensures that we finish all <<09421>> on line processing before we exit. <<09421>>
- LW - Last operation is a WRITE. <<J9456>>
- MP - Must Powerfail the next operation. <<J9456>>
- SUBSTATE - Idle channel program state.
  - 0 - Normal idle channel program wait.
  - 1 - Idle request being serviced wait.

DBADBLK1 - High order logical sector address of bad block encountered.

DBADBLK2 - Low order logical sector address of bad block encountered.

DBADXFER - Byte transfer left when bad block occured.

DLOGERROR - CPVA (0) logged on hardware error status.

DSIOPSTOP - Relative offset location of channel program when error in CPVA (0) occured.

DBYTECNT - Accummulative transfer count for transfers greater than 6144 bytes.

DSTATUS - 20 bytes of status logged when a status error occurs. (Refer to CS/80 Instruction Set manual for description.)

Card Reader DIT

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
X0	0	0	ACT	REQ	0	0		I/O PROG		IRK	READ DONE	NR MSG		MSTATE			DFLAG
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:

BIT 0 = SIO OK
BIT 1 = 0
BIT 2 = Interrupt pending
BIT 3 = Timing error
BIT 4 = Light dark check
BITS 5-6 = 00 Column binary mode
           01 Unused
           10 Packed binary mode
           11 Hollerith-to-ASCII mode
BIT 7 = Compare error
BIT 8 = EOF detected
BITS 9-10 = 00 Normal
            01 Hopper empty
            10 Unused
            11 Stacker full
BIT 11 = Invalid Hollerith
BIT 12 = Pick fail or motor check
BIT 13 = Test
BIT 14 = Trouble
BIT 15 = 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.
- IRK - 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.

UNIT - Unit number of device.

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 it 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.

DUNIT - I/O system type and unit number.

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
        |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
    X0| 0| 0|AC|RQ| 0|MU| 0|IO|IA|NO|ST| 0|   STATE   |DFLAG
        |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
     1|       SYSDB RELATIVE POINTER TO THE DIT         |DLINK
        |          FOR THE NEXT DEVICE REQUESTING       |
        |          THIS RESOURCE OR SERVICE             |
        |-----------------------------------------------|
     2|       IOQ TABLE RELATIVE INDEX TO THE FIRST     |DIOQP
        |          IOQ IN THE REQUEST LIST FOR THIS     |
        |          DEVICE                               |
        |-----------------------------------------------|
     3|       LOGICAL DEVICE NUMBER                     |DLDEV
        |-----------------------------------------------|
     4|       SYSDB RELATIVE POINTER TO DEVICE          |DDLTP
        |          LINKAGE TABLE                        |
        |-----------------------------------------------|
     5|       SYSDB RELATIVE POINTER TO THE             |DILTP
        |          INTERRUPT LINKAGE TABLE              |
        |--|--|-----------------------------------------|
     6|RD|RF|                                           |DSAVE
        |--|--|-----------------------------------------|
     7|       HARDWARE ERROR STATUS                     |DSERR
        |          SET WHEN THE DRIVER 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          |DSTAT
        |          DURING 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 IRK   - An interrupt or response has occurred for this
              device.
NO NOTRDY - Go to state X10 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:
                X0 - 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.

DUNIT - I/O system type and unit 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.
RF 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
        |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
    X0| 0| 0|AC|RQ| 0| 0| 0|IO|IA|NO|ST| 0|   STATE   |DFLAG
        |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
     1|       SYSDB RELATIVE POINTER TO THE DIT         |DLINK
        |          FOR THE NEXT DEVICE REQUESTING       |
        |          THIS RESOURCE OR SERVICE             |
        |-----------------------------------------------|
     2|       IOQ TABLE RELATIVE INDEX TO THE FIRST IOQ |DIOQP
        |          IOQ IN THE REQUEST LIST FOR          |
        |          THIS DEVICE                          |
        |-----------------------------------------------|
     3|       LOGICAL DEVICE NUMBER                     |DLDEV
        |-----------------------------------------------|
     4|       SYSDB RELATIVE POINTER TO THE             |DDLTP
        |          DEVICE LINKAGE TABLE                 |
        |-----------------------------------------------|
     5|       SYSDB RELATIVE POINTER TO THE             |DILTP
        |          INTERRUPT LINKAGE TABLE              |
        |--|--|----------------------------|--|--|--|---|
     6|VM|   |    TAB                      |PS|FL|TP|   |DSAVE
        |--|--|----------------------------|--|--|--|---|
     7|       HARDWARE ERROR POINTER                    |DSERR
        |          SET WHEN THE DRIVER 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     |DRQST
        |          WHILE 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 IRK   - An interrupt or response has occurred for this device.
NO NOTRDY - Go to state X10 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:
                X0 - 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.

DUNIT - I/O system type and unit 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:
VM VFCMOD   - VFC has been modified.
TAB TABDFAULT - System tab default.
PS PRESFACE - 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 mechanism error

 10     Self test failure
 11     Paper error
 12     Self test mode

 13     6/8 lines per inch
 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 HP 2631
controller.) The following diagram shows the DIT used for the
HP 2631 line printer driver.

```
    0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15  MNEMONIC
   |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
 X0| 0| 0|AC|RQ| 0| 0| 0|IO|IA|NO|ST| 0|   STATE   |DFLAG
   |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
  1|   SYSDB RELATIVE POINTER TO THE DIT            |DLINK
   |   FOR THE NEXT DEVICE REQUESTING               |
   |   THIS RESOURCE OR SERVICE                      |
   |------------------------------------------------|
  2|   IOQ TABLE RELATIVE INDEX TO THE FIRST         |DIOQP
   |   IOQ IN THE REQUEST LIST FOR                   |
   |   THIS DEVICE                                   |
   |------------------------------------------------|
  3|   LOGICAL DEVICE NUMBER                         |DLDEV
   |------------------------------------------------|
  4|   SYSDB RELATIVE POINTER TO THE                 |DDLTP
   |   DEVICE LINKAGE TABLE                          |
   |------------------------------------------------|
  5|   SYSDB RELATIVE POINTER TO THE                 |DILTP
   |   INTERRUPT LINKAGE TABLE                       |
   |--------------------------------|--|--|--|--|--|-|
  6|                                |BJ|RB|PS|FL|TP|DSAVE
   |--------------------------------|--|--|--|--|--|-|
  7|   HARDWARE ERROR STATUS                         |DSERR
   |   SET WHEN THE DRIVER 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        |DRQST
   |   WHILE 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 IRK   - An interrupt or response has occurred for this device.
  NO NOTRDY - Go to state X10 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:
        X0 - 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 nake present, then state 2.
        17 - Wait for completor nake present, then state 3.

DUNIT - I/O system type and unit 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:
  BJ BETJOB   - Between jobs flag: if set, the Powerfail
                 message is suppressed.
  RB 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
    |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
 DIT 0|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                  |CIOQP
    |-----------------------------------------------|
   3|   LOGICAL DEVICE NUMBER                        |DLDEV
    |-----------------------------------------------|
   4|   DRIVER LINKAGE TABLE POINTER                 |DDLTP
    |-----------------------------------------------|
   5|   INTERRUPT LINKAGE TABLE POINTER              |DILTP
    |-----------------------------------------------|
   6|   SPECIAL ERROR CONDITIONS TO BE LOGGED        |CSTAT
    |-----------------------------------------------|
   7|   ERROR LOGGING INFORMATION                    |DSERR
    |--|--------------------------------------------|
  10|T |   TIMEOUT INDICATION IN BIT 0               |DTIME
    |--|--------------------------------------------|
  11|   TIMER REQUEST INDEX (TRL) OR ZERO            |DTRLX
    |-----|-----------------------------------------|
  12| IOT |                PHYSICAL UNIT #           |DUNIT
    |-----|-----------------------------------------|
  13|   CURRENT DATA WRITE BYTE COUNT                |DCBCNT
    |-----------------------------------------------|
  14|   CURRENT DATA WORD COUNT                      |DCWCNT
    |-----------------------------------------------|
  15|   # OF WORDS LEFT TO TRANSFER                  |DRCNT
    |-----------------------------------------------|
  16|   BUFFER OFFSET FOR NEXT # OF WORDS TO XFER    |DOFFSET
    |-----------------------------------------------|
  17|                                               |DDDEBUG
    |-----------------------------------------------|
  20|   I/O STATUS BLOCK WORD 1 GETS                 |DLOGBUFFER
    |   LOGGED FROM HERE                             |
    |-----------------------------------------------|
  21|   I/O STATUS BLOCK WORD 3 GETS                 |
    |   LOGGED FROM HERE                             |
    |-----------------------------------------------|
  22|   I/O STATUS AREA                              |DIOSTAT
    |   (16 WORDS, SEE DEFINITION)                   |
    |-----------------------------------------------|
```

DFLAG - Device relative flags:
        AC - Active bit - 1 implies that a monitor is currently
             servicing this device.
        RQ - Request bit - 1 implies service requested while the
             monitor is active.

SP - SIO preemption - if set, then a preemptive request
    has been queued for this device; the preempt code is
    set in the 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, the device is in a not ready or operator wait
    state.
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:
        IO - Start request.
         1 - Not used (reserved).
         2 - Call driver initiator.
         3 - Call driver completor.
         4 - Unused (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/freeze).
        12 - Waiting (initiator code makepresent/freeze).
        13 - Waiting (for completion interrupt).
        14 - Waiting (for device controller availability).
        15 - Unused (reserved).
        16 - Waiting (initiator code makepresent).
        17 - Waiting (completor code makepresent).

DUNIT - I/O system type and unit number.
     IOT - I/O system type:
        0 - HP 3000 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 be transferred.

DOFFSET - Offset in buffer of next number of words to transfer.

DDEBUG - If bit 15 = 1 then debugging information will be sent
     to the 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.

INP Device Information Table (DIT)

	0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15	
DIT0	0 |AC|RQ|TI| 0|PR|IO|IN|SM|MAMSTATE| IOSTATE	DFLAG
1	POINTER TO NEXT DIT	DLINK
2	INPUT REQUEST QUEUE	DIOQX
3	LOGICAL DEVICE NUMBER	DLDEV
4	DRIVER LINKAGE TABLE POINTER	DDLTP
5	INTERRUPT LINKAGE TABLE POINTER	DILTP
6	INTERRUPT STATUS	DSTATUS
7	SOFTWARE TIMER REQUEST INDEX	DTRLX
10	SW|HW| HARDWARE TIMER REQUEST INDEX (33)	DTIME
11	RESERVED	
12	RESERVED	
13	READY QUEUE HEAD POINTER	READYQ
14	READY QUEUE TAIL POINTER	
15	ACTIVE QUEUE HEAD POINTER	ACTIVEQ
16	ACTIVE QUEUE TAIL POINTER	
17	WAITED QUEUE HEAD POINTER	WAITEDQ
20	WAITED QUEUE TAIL POINTER	
21	EO|WP|TR| PFSTATE |UF|PR|NR|SD| OS |RB	DSTATE
22	RESERVED | MESSAGE TO INP TYPE	DOUTMSG
23	REQUEST IDENTIFIER (IOQX)	DOUTID
24	PARAMETER 1 (QMISC)	DOUTP1
25	OUT COUNT	DOUTCNT

INP Device Information Table (DIT) (Cont.)

	0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15	
26	PARAMETER 2 (QPAR2)	DOUTP2
27	SEND DIALOG COUNTER	DSEND
30	RECEIVE DIALOG COUNTER	DRECV
31	"MESSAGE SENT" ECT BUFFER	DEOT
32	RESERVED | MESSAGE FROM INP TYPE	DINMSG
33	REQUEST IDENTIFIER (IOC X)	DINID
34	ERROR CODE - |WT| |CN| STATUS	DRSTATUS
35	IN COUNT	DINCNT
36	TRANSMISSION LOG	DXLOG
37	PARAMETER	DINPARM
40	TRACE READY REQUESTS COUNT	DTRCNT
41	EXTERNAL TRACE EXTRA DATA SEGMENT NUMBER	DDSTN
42	RESERVED | OUT MSG TYPE AT ERROR	DERROR
43	REQUEST IDENTIFIER (IOQX)	
44	PARAMETER 1 (QMISC)	
45	OUT COUNT	
46	PARAMETER 2 (QPAR2)	
47	LAST CS ERROR CODE	DCSERR
50	IOQP POINTER AT TIME OF ERROR	DSAVE
51	TP|PHY DRVR VERSN # |M| LOGICAL DRVR VERSN #	DVERSION
52	RESERVED | IN MSG TYPE AT ERROR	DERROR1
53	REQUEST IDENTIFIER (IOQX)	
54	ERROR CODE |WT| |CN| STATUS	
55	IN COUNT	

INP Device Information Table (DIT) (Cont.)

	0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15	
56	TRANSMISSION LOG	
57	PARAMETER	
60	DRIVER ERROR CODE	DDRVRERR
61	MONITOR ERROR CODE	DMNTRERR
62	HARDWARE ERROR STATUS | SIO PROGRAM INDEX	DSERR
63	TOOTHPICK HARDWARE ERROR STATUS	DTP'ERROR
64		
65	DRIVER TRACE READ IOQ INDEX	DTR'IOQX
66	RESERVED	
67	DSTN FOR PORT TRANSLATOR	DTRANDSTN
70	PLABEL FOR PORT TRANSLATOR	DTRANPLBL
71	INP CONTROLLER DIT SIZE	DITSIZE

INP DIT Field Definitions:

DFLAG   - Flags, IOSTATE and MAMSTATE.

.ACTIVE - If set, the Driver is active servicing this device.

.REQUEST - If set, service for this device was requested while the
         Driver was active. The Driver is run again to insure
         servicing of the condition which caused REQUEST to be set.

.DO'TIMING - If set, the hardware and software timers are started in
         the normal manner when performing an operation. If clear,
         no timing is done.

.SIOPREEMPT- Preemptive request queued by ATTACHIO. Not used by this
         Driver.

.IOPROG - If set, an I/O program is in progress. Set by STARTIO and
         cleared by GIP. Not used by the Driver.

.IAK   - Interrupt Acknowledge. If set, an interrupt has occurred or
         a software timeout has completed.

.SIMULATOR - If set, all I/O is to be simulated. The Driver will set flags in the DRT instead of calling STARTIO.

.MAMSTATE - Memory Manager State.
0 - Null, no Memory Management requests or condition.
1 - Not used.
2 - Data segment associated with the first request in the Active Queue is being made present and frozen.
3 - Data segment associated with the first request in the Active Queue is frozen in memory.
4 - Data segment associated with the second request in the Active Queue is being made present and frozen. Implies the data segment associated with the first request is frozen.
5 - Data segments associated with the first and second requests on the Active Queue are frozen in memory.
6 - Not used.
7 - Not used.

.IOSTATE - Current I/O program operation being performed.
0 - Inactive. No I/O in progress.
1 - Idle Read. The Idle Read I/O program has been started.
2 - Sending message. An I/O program which sends a message without data and then goes to the Idle Read section of the I/O program has been started.
3 - Sending data. An I/O program which sends a message and data and then goes to the Idle Read section has been started.
4 - Send message and interrupt. An I/O program which sends a message without data then interrupts and halts when the message sent has been started.
5 - Send data and interrupt. An I/O program which sends a message with data then interrupts and halts has been started.
6 - Receive data. An I/O program which sends a message and receives data then interrupts and halts has been started.
7 - Do not start I/O. Used to hold off requesting any I/O activity during a power on reset or when an error occurs.

DLINK - Link word for the linked list of devices waiting to be served by the I/O process associated with this device.

DIOQP - System DB relative pointer to the first element in the request to be processed list for this device. The requests are queued to this list by ATTACHIO but in processing, they are moved to other queues depending of the state of the request. the Driver always attempts to keep this list empty.

DLDEV - Logical Device Number of this device.

DDLTP - System DB relative pointer to the Driver Linkage Table (DLT)

DILTP - System DB relative pointer to the Interrupt Linkage Table (ILT)

DSTATUS - Controller hardware status. Set by GIP on interrupt and the Physical Driver during certain service operations. See INP ERS for description. For the Toothpick version, this word contains the software timeout flags as described for the word DTIME below.

DTRLX - Timer request index for software timeouts as returned by the MPE procedure TIMEREQ.

DTIME - Timed out flags and type 3 driver process PCB Number.

.TIMED - If set, a software timeout has completed.

READYQ - System DB relative pointer to the IOQ for the first request the Ready Queue. If zero, the Ready Queue is empty.

READYQTL - System DB relative pointer to the last IOQ in the Ready Queue. When the queue is empty, this word points to the word preceding then queue head pointer in the DIT.

ACTIVEQ - System DB relative pointer to the IOQ for the first request the Active Queue. If zero, the Active Queue is empty.

ACTIVEQTL - System DB relative pointer to the last IOQ in the Active Queue. When the queue is empty, this word points to the word preceding then queue head pointer in the DIT.

WAITEDQ - System DB relative pointer to the IOQ for the first request the Waited Queue. If zero, the Waited Queue is empty.

WAITEDQTL - System DB relative pointer to the last IOQ in the Waited Queue. When the queue is empty, this word points to the word preceding then queue head pointer in the DIT.

DSTATE - Driver state and control flags.

.ERRORONLY - If set, the Driver trace record is to be returned to the Trace Process only when an error occurs.

.WRAP - If set, the Driver will overlay the oldest trace entry when a trace record overflow occurs. If clear, entries are lost when an overflow occurs.

.TRACEON - If set, the Driver trace facility is enabled and the Driver generates trace entries for most of its local subroutine calls.

.PFSTATE - Power failure recovery state.
0 - No power failure recovery in progress.
1 - Powerfailure detected on the Mainframe before INP indication. Check for completion of any pending I/O and then wait in PFSTATE 2 for INP to PFAIL.

---

2 - Power failure detected on the Mainframe before INP has indicated a power failure. Wait for INP to indicate a power failure.
3 - Power failure indicated by INP before being informed by the Mainframe power failure routines. Wait for the Mainframe power failed request.
4 - Power failure indicated both on the Mainframe and by INP. Power failure recovery may be started.
5 - Send Redo. The Mainframe receive count was less than the INP send count so the dialog must be restarted. The Driver is sending the Redo message.
6 - Send Ignore. The Mainframe send count was greater than the INP receive count so any part of a dialog so far received is to be ignored and the entire dialog will be retransmitted. The Driver is sending Ignore message
7 - Recovered. The Mainframe and INP dialog counters agree or Mainframe not sending, so no recovery is necessary. The Driver is sending the recovered message informing INP to go back to its normal mode.

.UNFRZ - If set, the source data segment is to be unfrozen when the data has been transmitted to the INP. If clear, the source data segment remains frozen until a request complete indication is returned by the INP.

.PASSREADS - If set, then read requests are to be passed around other requests which have been impeded because no buffers are available on the INP.

.NOTRDYWAIT- If set, then a request has been impeded because no buffers were available on the INP.

.SENDING - If set, an I/O program which sends a message, with or without associated data, has been started but not completed.

.OPENSTATE - Operational state of the Driver and INP.
0 - Not opened or closed.
1 - In ROM. The device has been opened but the RAM Operating System has not been entered.
2 - Crashed. Some catastrophic error has occurred.
3 - In RAM. The device has been opened, down loaded, and is in the RAM Operating System.

.ABORT - If set, one or more requests have been aborted but the abort was not done because the aborted request was in the process of doing a Memory Management function or I/O when request to abort was processed. The actual abort will take place when the Memory Management function completes.

DOUTMSG - Message type code for messages sent to INP.

DOUTID - Request identifier associated with the message being sent.

DOUTP1 - Parameter one of the message being sent to INP.

DOUTCNT - Count parameter of the message being sent to INP.

DOUTP2 - Parameter two of the message being sent to INP.

DSEND - Messages sent counter. This word contains the number of messages sent since the RAM Operating System was entered. It is used for power failure recovery.

DRECV - Messages received counter. This word contains the number of messages received from INP since the RAM Operating System was entered. It is used for power failure recovery.

DEOT - End of dialog flag. When a message has been sent and the EOT indicating INP has received the message is transmitted, is received into this word. This flag is used to indicate the Logical Driver that a transmission has been completed and the Physical Driver should be called to check the completion status and update the IOSTATE.

DINMSG - Message type code of message from INP.

DINID - Request Identifier associated with message from INP.

DRSTATUS - Request Completion status.

DINCNT - Number of bytes of data to be received associated with the completion of a request which results in data being sent from INP.

DXLOG - Transmission log to be returned when the request identified by DINID is completed.

DINPARM - Parameter associated with the completion of this request. This word is returned in the X register by IOSTATUSX.

DTRCNT - Trace ready pending count. This word contains the number of Trace Ready messages received but not satisfied by Trace Ready requests.

DDSTN - If not zero, then internal Driver extra data segment tracing is enabled and this is the data segment number into which the trace entries are to be set.

DERROR - Driver Error block. The following sixteen words are used to store information describing the current operations being performed when a catastrophic Driver error occurred. A catastrophic error occurs on illogical Driver control data, MPE errors, or when INP does not respond in an expected manner. The first five-word block is used to hold the current or last message transmitted to INP when a catastrophic error condition was detected. It contains the data in the same form as message to INP block.

DCSERR   - CS Error Code associated with a catastrophic Driver error.

DSAVE   - Request Identifier of the request being processed when a
            catastrophic Driver error was detected.

DVERSION - Version numbers of the Physical and Logical Drivers.

.TP   - If set, the Physical Driver is for the Toothpick System.

.PVERSION - Physical Driver version number.

.LVERSION - Logical Driver version number.

DERROR1   - The six-word block beginning here is used to hold the last
            message received from INP before a catastrophic Driver error
            was detected. It contains the data in the same format as the
            message from INP block.

DDRVRERR - Holds the code specifying the catastrophic error detected
            by the Physical Driver. See ERRORS under the PHYSICAL DRIVER
            INTERNAL SPECIFICATIONS for the definition.

DMNTRERR - Holds the code specifying the catastrophic error detected
            by the Logical Driver. See ERRORS under the LOGICAL DRIVER
            INTERNAL SPECIFICATIONS for the definition.

DSERR   - Hardware Controller status when a catastrophic Driver error
            was detected.

.HSTATUS - Left byte of the DSTATUS word at time of error.

.SIOPX   - SIO program area relative index to the last order executed
            or current order being executed at time of error.

DTP'ERROR - Toothpick hardware error status. To be defined.

DTR'IOQX - If not zero, then an IOQP pointer to the Trace Read request
            which is supplying the locked and frozen buffer into which
            the Driver places trace entries to generate a trace record.

DTRANDSTN - DSTN for port translator.

DTRANPLBL - PLABEL for the port translator.

DITSIZE   - INP controller DIT size.

I/O Status Block

```
      0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15  DIT
     |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
   0 |0 |  THE "OR" OF WORDS 1/15 IS LOCATED HERE    |%21
     |--|--|--|--|--|----------------------------------|
   1 |OF|MS|PU|PE|TE|                                  |%22
     |--|--|--|--|--|----------------------------------|
   2 |            RESERVED                             |%23
     |------------------------------------------------|
   3 |       MCS FAULT NUMBER                          |%24
     |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
   4 |CL|FL|VL|CU|FU|VU|IL|IP|ST|SB|IR|MP|NJ|NM|TL|NC|%25
     |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
   5 |LP|PF|NC|        RESERVED                        |%26
     |--|--|--|----------------------------------------|
   6 |            RESERVED                             |%27
     |------------------------------------------------|
   7 |            RESERVED                             |%30
     |------------------------------------------------|
  10 |            RESERVED                             |%31
     |------------------------------------------------|
  11 |            RESERVED                             |%32
     |------------------------------------------------|
  12 |            RESERVED                             |%33
     |------------------------------------------------|
  13 |            RESERVED                             |%34
     |------------------------------------------------|
  14 |     RECORD NUMBER OF ERROR                      |%35
     |-         IF WORD 4 IS                          -|
  15 |          NON-ZERO                               |%36
     |------------------------------------------------|
  16 |     SHEET NUMBER OF ERROR IF WORD 4 IS          |%37
     |-    NON-ZERO OR LAST SHEET TRANSFERRED         -|
  17 |     IF "JOB" AND POWER ARE ON                   |%40
     |------------------------------------------------|
```

WORD 0   - Each bit is the 'OR' of one word in the table (except
            bit 0 which is not used); 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 - (PU) 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   - Reserved (unused).

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
                       the 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, VFCs,
                       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.

WORD 6/13 - Reserved for future use (unused).

WORD 14/15 - 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).

WORD 16/17 - 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;
             additionally, 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 a "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 - Miscellaneous request dependent storage available to driver.

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

    .(0:1) - (MB) user requested a 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 times out.

    .(5:4) - Reserved (unused).

    .(9:3) - (XFER) 2680A/2688A transfer error counter.

    .(12:3) - (PARITY) channel program command parity error counter.

    .(15:1) - Reserved (unused).

    NOTEIn the above example, single bit fields are as defined
            when the bit is a logical "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 queue.

The head and tail of a device's request queue are contained in the devices' DIT.

DISCREQTAB

```
                          |------------|
                          |            |
                /-------->|------------|
                |         |  DEVICE'S  |
|------|        |  /------|  CURRENT   |
|      |---/    |  |      |  REQUEST   |
| DIT  |-------\|  |      |------------|
|      |---\    \  |         |
|------|    |   |  |         |
           |   |  |         |
           |   |  |         |
           |   |  |         |
           |   \-->|------------|
           |      | FIRST      |
           |      | REQUEST    |
           |      | IN QUEUE   |<>----\
           |      |------------|      |
           |         |                |
           |         |                |
           |         |                |
           |      |------------|      |
           |      | NEXT       |<>----/
           |      | REQUEST    |
           |      | IN QUEUE   |<>----\
           |      |------------|      |
           |         |                |
           |         |                |
           |      |------------|      |
           \----->| LAST       |<>----/
                  | REQUEST    |
                  | IN QUEUE   |
                  |------------|
                     |
                     |
                  |------------|
```
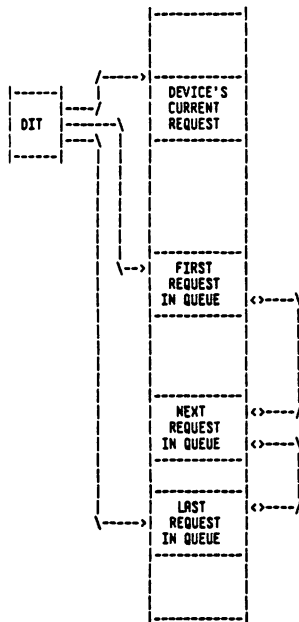
G.23.00
13- 87

Disc Request Table

```
|   DISCREQTAB DST = 56 (Z70)
    DISCREQTAB PRT = Z1017
```

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 (Z21)	
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	
DISCREQTAB10	OVERFLOWS	
DISCREQTAB11	TOTAL REQUESTS	
DISCREQTAB12		
DISCREQTAB13	SYSBASE INDEX OF HEAD OF DISABLED REQ Q	DISCQHEAD
DISCREQTAB14	SYSBASE INDEX OF TAIL OF DISABLED REQ Q	DISCQTAIL
DISCREQTAB15	SERIAL WRITE QUEUE HEAD	SERWQHEAD
DISCREQTAB16	A | MAX. SERIAL WRITE QUEUE	A = Active
DISCREQTAB17		
DISCREQTAB20		

G.23.00
13- 88

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 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 I D M T E O A R S R L / R E G F A P A I U F I R A O / T Q K E I F R E A L E B C / E L I Q A / I 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 10	P1 (KDDA IF SEGMENT TRANSFER)	PAR1
Word 11	P2 (LDDA IF SEGMENT TRANSFER)	PAR2
Word 12	QUALIFIER | STATUS	STAT
Word 13	FR | PCB NUMBER	PCBN
Word 14	INDEX OF PREV REQUEST IN QUEUE	PREVREQP
Word 15	INDEX OF NEXT REQUEST IN QUEUE	NEXTREQP
Word 16		
Word 17	SEGIDENTIFIER (IF SEG TRANSFER)	SEGIDENT
Word 20	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.

G.23.00
13- 89

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 awoken
                           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 request
                           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 logical 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 instead of segment relative
        offset (implemented for NOWAIT IO and NOBUF).

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.

G.23.00
13- 90

Word 7 - QXFERCNT - On initiation specifies the word count if positive
or the byte count if negative. At completion of the request
this location contains the actual transmission count in the
same units as the original call. Certain control requests
return data through this location.

Word 10 - QPAR1 - Parameter one, defined by driver.

Word 11 - QPAR2 - Parameter two, defined by driver.

Word 12 -
QSTAT.QUALIFIER - A code which further defies or qualifies the
general status. Defined by the driver.
QSTAT.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 13 - QPCBN.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.

Word 13 - bit 0 = 1 - Q element is on free list.

NOTE: See I/O System Status Returns later in this chapter.

I/O Queue (IOQ) Table Layout

I/O Queue (IOQ) Table Layout (Cont.)

I/O Queue Element (IOQ)

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 awoken
if he had specified.
Bit 7   .DATAFRZN - Data segment has been made present and is
frozen.
Bit 8 .MAMERRORPD - MAM error on data segment make present.
Bit 9   .PREQ - This request has been started but was
preempted by a MAM request.

I/O Queue Element (Cont.)

Bit 10   .SFAIL - Start SIO failure in GIP.
Bit 11   .PFAIL - The I/O has been aborted because of a
              powerfail.
Bit 12/13 .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 - Miscellaneous request dependent storage available to driver.

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) - S(Word 4(0:1) - Stackflag - If set
        is DB relative.

QRDDR - Offset in data segment or system buffer 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.

QSTAT - .QUALIFIER - A code which further defies or qualifies the
                 general status; defined by the driver.
QSTAT - .STATUS - General Status. Indicates the current and resulting
                 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.

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.

Word 13 bit 0 - Queue element is on free list.

G.23.00
13- 95

I/O System Status Returns

		STATUS X
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 online	53
	6 - Aborted because of power fail	63
	7 - EOT 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

G.23.00
13- 96

I/O System Status Returns (Cont.)

		STATUS X
4 - Irrecoverable Error		
	0 - Invalid request	4
	1 - Transmission error	14
	2 - I/O timeout	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 the 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 reel switch	204
	21 - Parity error detected on PHI command (EPOC)	214

		STATUS X	XLOG
5 - Error In Data Control Information			
	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 information file format	45	
	5 - Checksum error in information file	55	
	6 - Passed value less than minimum	65	MIN.VALUE ALLOWED
	7 - Passed value greater than maximum	75	MAX.VALUE ALLOWED
	10 - Passed value is unsupported	105	
	11 - Count less than required to return all information	115	MIN.SPACE NEEDED
	12 - Count greater than available for storing information	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

G.23.00
13- 97

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
    |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
  X0|          REQUEST DEPENDENT FLAGS          | QFLAG
 |  |------------------------------------------|
  1|   SYSDB RELATIVE POINTER TO THE  IOQ       | QLINK
    |   NEXT IOQ ELEMENT - POINTS TO THE        |
    |   FIRST WORD OF THE ELEMENT               |
 |  |------------------------------------------|
  2|          LOGICAL DEVICE NUMBER             | QLDEV
 |  |--|--|--|--|--|----|-------|-------|-------|
  3| R| B| F| G|B0| TOUT| FSCNTR| BSCNTR| RTCNTR| QMISC
    |--|--|--|--|--|----|-------|-------|-------|
  4| S|                                         | QDSTN
    |--|---------------------------------------|
  5|    OFFSET IN THE DATA SEGMENT OR SYSTEM    | QRDDR
    |    BUFFER TABLE TO THE TARGET DATA        |
    |    BUFFER                                 |
    |-----------------|------------------------|
  6|                  |       FUNCTION CODE     | QFUNC
    |-----------------|------------------------|
  7|                                           | QWBCT
    |------------------------------------------|
 10|    PARAMETER 1 - USED ONLY FOR READS       | QPAR1
    |    CONTAINS THE EOF SPECIFICATION         |
    |    IN BITS (13:3)                         |
    |------------------------------------------|
 11|    PARAMETER 2 - USED ONLY FOR WRITES      | QPAR2
    |    IF BIT (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 (unused).
  Bit 2 DIAG   - This is a request from the diagnostic subsystem
                    (unused).
  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 (MRM) when a
                    MAKEPRESENT request is successfully completed and
                    indicates the data segment is frozen in memory.

G.23.00
13- 98

Bit 8 MAKERRORD - An error has occurred while MAM was trying to
             make the target data segment present and freeze
             it in memory.
Bit 9 PREQ   - Unused.
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.

QDSTN - If QFLAG.(3:1) is clear then this is the DST number of the
     target data segment - if S is set, QADDR is DB relative.

QMISC - Driver request dependant flags and counters - used mostly for
     error retries.
     R   - Indicates an error retry is in progress.
     B   - Backspace record processing for an error retry is in
           progress.
     F   - Forward space record processing for an error retry is
           in progress.
     G   - Gap processing for an error retry is in progress.
     BO   - Backspace record due to a data EOF processing is in
           progress.
     TOUT   - GIC timed-out counter.
     FSCNTR - Forward space record counter.
     BSCNTR - Backspace record counter.
     RTCNTR - Error retry counter.

QWBCT - On initiation, specifies the word count (> 0) 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.

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 on the next page).

I/O Queue Element (IOQ) for CIPER

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 (unused).
     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 MAKERRORD - An error has occurred while MAM was trying to
                make the target data segment present and freeze
                it in memory.

Bit 9 PREQ   - (Unused).
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.

QDSTN - If QFLAG.(3:1) is clear then this is the DST number of the
     target data segment; if S is set, QADDR is DB relative.

QWBCT - On initiation, specifies the word count (> 0) 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.

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
           on the next page).

MP-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	(Unused)	
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
     |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
  X0|        REQUEST DEPENDENT FLAGS          |QFLAG
    |----------------------------------------|
   1|  SYSDB RELATIVE POINTER TO THE NEXT IOQ |QLINK
    |  ELEMENT - POINTS TO FIRST ELEMENT      |
    |----------------------------------------|
   2|         LOGICAL DEVICE NUMBER           |QLDEV
    |--|--|--|---------|-----------|----------|
   3|PP|PE|MC|TOUTCNTR |           |  WAITCODE |QMISC
    |--|--------------------------------------|
   4| S|                                      |QDSTN
    |--|-------------------------------------|
   5|   OFFSET IN THE DATA SEGMENT OR SYSTEM  |QADDR
    |   BUFFER TABLE TO THE TARGET DATA       |
    |   BUFFER                                |
    |----------------------------------------|
   6|                    | FUNCTION CODE FOR  |QFUNC
    |                    |   THIS REQUEST     |
    |----------------------------------------|
   7|                                         |QWBCT
    |----------------------------------------|
  10|          PARAMETER 1                    |QPAR1
    |----------------------------------------|
  11|          PARAMETER 2                    |QPAR2
    |----------------------------------------|
  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 (unused).
  Bit 2 DIAG   - This is a request from the diagnostic subsystem
                     (unused).
  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   - (Unused).
  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, or
                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.

QDSTN - If QFLAG.(3:1) is clear then this is the DST number of the
        target data segment; if S is set, QADDR is DB relative.

QWBCT - On initiation, specifies the word count (> 0) 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.

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
              above).

2608 Line Printer Request Codes

Operation	Function	Parameters
WRITE	1	P1 - Vertical Format Specification. 1 - Use 1st data char as format spec. X53 - "+", print and suppress spacing. X55 - "-", print and triple space. X60 - "0", print and double space. X61 - "1", print and top of form. X200-X277 - Print and space n-X200 lines. X300-X377 - Print with channel n-X277. 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	X17	Read I/O status. Count - buffer must be at least 2 bytes.
VFC SET	X100	Load VFC RAM. Count - Form length in words (0 loads RAM from internal ROM). P1 - 6 for 6 LPI or 8 for 8 LPI any other value defaults to 6 LPI.
TAB SET	X101	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
     |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
  X0|        REQUEST DEPENDENT FLAGS          |QFLAG
    |----------------------------------------|
   1|  SYSDB RELATIVE POINTER TO NEXT IOQ     |QLINK
    |  ELEMENT - POINTS TO FIRST WORD         |
    |  OF ELEMENT                             |
    |----------------------------------------|
   2|         LOGICAL DEVICE NUMBER           |QLDEV
    |--|--|--|---------|-----------|----------|
   3|PP|PE|PF|TOUTCNTR |           |  WAITCODE |QMISC
    |--|-------------------------------------|
   4| S|                                      |QDSTN
    |--|-------------------------------------|
   5|   OFFSET IN THE DATA SEGMENT OR SYSTEM  |QADDR
    |   BUFFER TABLE TO THE TARGET DATA       |
    |   BUFFER                                |
    |----------------------------------------|
   6|                    | FUNCTION CODE      |QFUNC
    |----------------------------------------|
   7|                                         |QWBCT
    |----------------------------------------|
  10|          PARAMETER 1                    |QPAR1
    |----------------------------------------|
  11|          PARAMETER 2                    |QPAR2
    |----------------------------------------|
  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 (unused).
  Bit 2 DIAG   - This is a request from the diagnostic subsystem
                     (unused).
  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     - (Unused).
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.

QDSTN - If QFLAG.(3:1) is clear then this is the DST number of the
        target data segment; if S is set, QADDR is DB relative.

QWBCT - On initiation, specifies the word count (> 0) 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.

Format For 2619A

```
|   0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
|  |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
| Z3|PP|PE|PF|TO|BF|                    |WAITCODE|QMISC
|  |--|--|--|--|--|--------------------|--------|
|
|  |-----------------------------|-----|-----|
| 12|                            | QUALIFIER   |STAT |QSTAT
|  |-----------------------------|-----|-----|
```
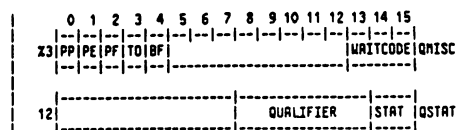
QMISC - Device dependent flags:
TOUTCNTR - (TO) Channel timeout flag.
BUF'FILL - (BF) Buffer fill operation in progress.

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.
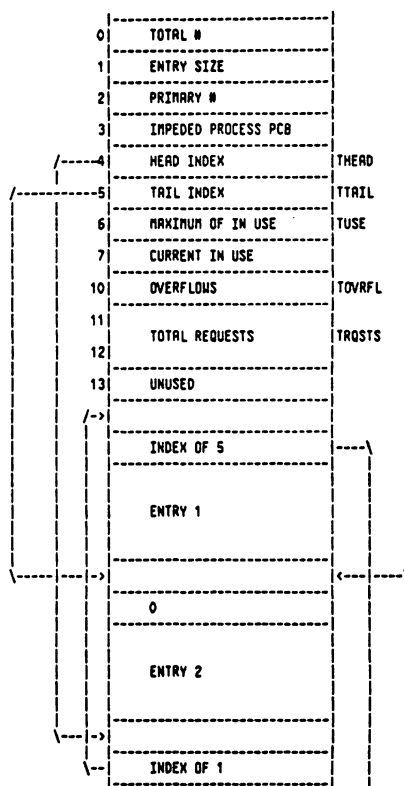   3 - Unusual, but recoverable, condition detected.
         4 - Irrecoverable error has occurred.
QUALIFIER - A code which further defines or qualifies the general
         status (see the section Driver Return Status Codes
         earlier in this chapter).

2619 Line Printer Request Codes

Operation	Function	Parameters
WRITE	1	P1 - Vertical Format Specification.

```
P1 - Vertical Format Specification.
     1 - Use 1st data char as format
         specification.

     X53 - "+", print and suppress spacing.
     X55 - "-", print and triple space.
     X60 - "0", print and double space.
     X61 - "1", print and top of form.

     X200-X277, Print and space n-X200 lines.
     X300-X312, Print with channel N-X277.

     X320 - 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).
```

Operation	Function	Parameters
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	X17	Read I/O status. Count - buffer size.
*IDENTIFY	X110	Return ID value in Bank & Buffaddr.

*SELF TEST:
```
INITIATE   X111    Subtest number to execute in Bank and Buffaddr
                   (subtest number ranges from 0 to 7).
STATUS     X112    Subtest result returned in Bank & Buffaddr.
```

*LOOPBACK TEST:
```
WRT DATA   X113    Data to LP in Bank & Buffaddr [PING].
READ DATA  X114    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.

```
P1 - Vertical Format Specification.
     1 - Use 1st data char as format
         specification.

     X53 - "+", print and suppress spacing.
     X55 - "-", print and triple space.
     X60 - "0", print and double space.
     X61 - "1", print and top of form.

     X200-X277, print and space N-X200 lines.
     X300-X307, print with channel N-X277.

     X320 - 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).
```

Operation	Function	Parameters
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	X17	Read I/O status. Count - 1 byte minimum required.
VFC SET	X100	LOADS VFC RAM

```
VFC SET    X100    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 target 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 earlier in this chapter.)

CS 80 Disc Request I/O 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
    |  |           OR                               |
    |  | DST (IF SEGMENT TRANSFER) [S=STACK]         |
    |--|--------------------------------------------|
  5|   OFFSET IN THE DATA SEG (IF PROCESS DISC I/O) | QADDR
    |              OR                               |
    |   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                                          | QPCB
    |-----------------------------------------------|
 14| SYSBASE RELATIVE INDX OF PREVIOUS REQ IN QUEUE | QPREVREQP
    |-----------------------------------------------|
 15| SYSBASE RELATIVE INDX OF NEXT REQ IN QUEUE     | QNEXTREQP
    |-----------------------------------------------|
 16|                                                | QSEGIDENT
    |- SEGIDENTIFIER (IF SEG TRANSFER)             --|
 17|                                                |
    |-----------------------------------------------|
 20|DISPLACEMENT OF READ OR WRITE FROM SEG BASE(MM) | QSEGDISP
    |-----------------------------------------------|
```

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.
IN'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.)

INP I/O Queue Element (IOQ)

```
          0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
         |--|--|--|--|--|--|--|--|--|--|--|-----|--|--|
   IOQ0 |AB| ID |SB|WA|BL|IC |FR|ER|RW|H |WC|PRMPT|TI|AR|QFLAG
         |--|--|--|--|--|--|--|--|--|--|--|-----|--|--|
   IOQ1 |   IOQ INDEX TO NEXT NEW REQUEST         |QLINK
         |----------------------------------------|
   IOQ2 |   LOGICAL DEVICE NUMBER                  |QLDEV
         |----------------------------------------|
   IOQ3 |   MISCELLANEOUS PARAMETER               |QMISC
         |--|-------------------------------------|
   IOQ4 |DB| DST NUMBER                           |QDSTN
         |--|-------------------------------------|
   IOQ5 |   TARGET DATA BUFFER ADDRESS            |QADDR
         |----------------------------------------|
   IOQ6 | QUEUE | SERVICE |  FUNCTION CODE        |QFUNC
         |----------------------------------------|
   IOQ7 |   WORD (+) OR BYTE (-) COUNT            |QWBCT
         |----------------------------------------|
   IOQ10|   PARAMETER 1/READ DATA COUNT           |QPAR1
         |----------------------------------------|
   IOQ11|   PARAMETER 2                           |QPAR2
         |---------------------|--|--|------------|
   IOQ12| ERROR CODE          |WR|LS|    |CSTATUS |QSTAT
         |---------------------|--|--|------------|
   IOQ13| PCB NUMBER                             |QPCBN
         |----------------------------------------|
```

INP IOQ Field Definitions

QFLAG - Flags and Control Information.

.ABORT - If set, then request has been aborted.

.DIAG - Diagnostic Flag. Not used.

.SYSBUFR - System Buffer Flag. Not used.

.IOWAKE - Wake caller on completion of request.

.BLOCKED - Blocked I/O. Caller is waited in ATTACHIO until the request
is completed. Implies wake.

.COMPLETED - Request has been completed and caller awoken (if specified)
and request is no longer known to the Driver.

.DATAFRZN - If set, the target data segment is frozen in memory.
Set by MAM when a delayed make present request is
successfully completed.

.MAMERRD - A MAM error has occurred in trying to make present and
freeze the target data segment.

.READWRITE - If set, then this request allows data to be received after
data is sent. The read target buffer offset is in QPAR1 and
the read target buffer length is in QPAR2.

.HELD - If set, processing of this request has been suspended
because INP did not have buffer space available.

.WORDCOUNT - If set, QWBCT specified words, else QWBCT specified
bytes.

.PREEMPT - Preempt Code. Not used.

.TIME - If set, a software timeout is started when the request
initiation message is sent to INP and the Request
Completion message must be received before the timeout
expires.

.ABORTER - If set, this is a request to abort another request.

QLINK - SYSDB relative pointer to the next new IOQ element.

QLDEV - Holds Logical Device Number and Current Queue Index.

.OLDEVN - Logical Device Number of Controller.

QMISC - Miscellaneous parameter. Use varies with Function Code.
See INP FUNCTIONS for specific meaning.

QDSTN - DST Number and Request State.

.DBFLAG - If set, QADDR is the offset from DB to the target
buffer, otherwise QADDR is the offset from the DST base.

QADDR - Offset to target data area from data segment base or DB.

QFUNC - Error Code and Function.

.QUEUE - DIT relative index to head of queue holding this requests.
0 - Input Queue.
9 - Ready Queue.
11 - Active Queue.
13 - Waited Queue.

.SERVICE - Service code. This field controls the operations to be done
for this request and its disposition on completion.

    0 - Send message only, no data.
    1 - Send message and data.
    2 - Move data from trace write to trace read buffer.
    3 - Move Logical Driver Status Block to target buffer.
    4 - This is a request to abort another request.
    5 - Message has been sent to INP.
    6 - Receive data from INP.
    7 - Issue a power on reset.
    8 - Complete request when IOSTATE is inactive.
    9 - Soft Abort pending on this request.
    10 - Send data requested with Soft Abort pending.
    11 - No service currently required for this request.

.FUNCTION - Function Code as specified by driver.

QWBCT - Word or byte count. May also be used to return information
certain functions. On initiation, it specifies a word count
positive or a byte count if negative. It is converted to a
count during preprocessing of the request with the sense kept
in the flag WORDCOUNT. At completion, the actual transmission
count is returned in this word with the same sense as the
original specification.

QPAR1 - Parameter one as defined by the driver. When
a request has been completed and data is to be received, the
word contains the byte count of the data to be received.

QPAR2 - Parameter two as defined by the driver.

QSTAT - Caller PCB Number and request completion status.

.ERRORCODE - The Irrecoverable Error Code as defined in CS ERS.

.LS - Line State. If set, the line is connected. This field is
valid only for read and write completions.

.WR - If set, this was a write request completion.

.CSTATUS - Encoded Completion Status.
    1 - Successful Completion.
    2 - End of Transmission.
    3 - Irrecoverable Error Completion.
    4 - Unrecovered Recoverable Error Completion.
    5 - Catastrophic Controller Error.

QPCBN - PCB Number of the originator of this request. If zero, this
    IOQ element is returned by the Logical Driver when the
    request is completed.

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|IN| RETRY   |LF|SP|     | WAITCODE   |QMISC
     |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
   4| S|  DST  (IF PROCESS DISC I/O)                  |QDSCTN
     |  |              OR                             |
     |  | DST (IF SEGMENT TRANSFER) [S=STACK]         |
     |--|---------------------------------------------|
   5| OFFSET IN THE DATA SEG (IF PROCESS DISC I/O)    |QADDR
     |              OR                                |
     | 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.
    IN'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
               earlier in this chapter.)

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                     <-----|
    |-----------------------------------|
```

G.23.00
13- 123

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.

G.23.00
13- 124

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 system	Impede	---
Ptape	Impede	---
Bad track	Primary	Forget request
IOQ's		
ATTACHIO (can be impeded)	Primary	Return IOQX-0
ATTACHIO (can be impeded)	Impede	---
SIODM (memory management)	Secondary	Sudden death
IOMESSAGE	Secondary	I/O error

HEADER DEFINITION:

Primary #	- Number of elements in the primary area.
Total #	- Total number of elements in the table.
Size	- Size in words of each element.
Impeded PCB	- If not zero then contains the PCB number of the first process waiting for an element in this table.
Head index	- Index of first free element.
Tail index	- Index of last free element.
In use	- Current number not in free list.
Overflows	- Number of requests made for an element.
Total requests	- Total number of elements requested.

G.23.00
13- 125

Interrupt Control Stack (ICS) Format

```
        0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
        |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
| QI-77|              RESERVED                |
        |----------------------------------------|
| QI-76|              RESERVED                |
        |----------------------------------------|
| QI-75|              RESERVED                |
        |----------------------------------------|
| QI-74|              RESERVED                |
        |----------------------------------------|
| QI-73|              RESERVED                |
        |----------------------------------------|
| QI-72|              RESERVED                |
        |----------------------------------------|
| QI-71|              RESERVED                |
        |----------------------------------------|
| QI-70|              RESERVED                |
        |----------------------------------------|
| QI-67|              RESERVED                |
        |----------------------------------------|
| QI-66|              RESERVED                |
        |----------------------------------------|
| QI-65|              RESERVED                |
        |----------------------------------------|
| QI-64|              RESERVED                |
        |----------------------------------------|
| QI-63|              RESERVED                |
        |----------------------------------------|
| QI-62|              RESERVED                |
        |----------------------------------------|
| QI-61| CANDIDATE PIN THAT SYSTEM IS SERVICING   | ICS'CANDPINCELL
        |----------------------------------------|
| QI-60| C FILTER LAST TRANSACTION TIME MULTIPLIER| ICS'CUTLASTTRANUTCELL
        |----------------------------------------|
| QI-57|              PAUSE                   | ICS'PAUSETIMECELL
        |-                                    -|
| QI-56|              TIME                    |
        |--|--|----------------------------------|
| QI-55|IN|IO| MEASUREMENT INTERFACE FLAGWORD | ICS'LISTSTATECELL
        |--|--|----------------------------------|
        | BACKGROUND FILTER USED FOR QUANTUM UPDATE | ICS'CUREFILTERCELL*
        |----------------------------------------|
| QI-53| BATCH FILTER USED FOR QUANTUM UPDATE | ICE'CURDFILTERCELL*
        |----------------------------------------|
| QI-52| C FILTER'S OLD C FILTER CALCULATION CONSTANT | ICS'CUTOLDFILTUTCELL
        |----------------------------------------|
| QI-51| C FILTER CALCULATION DIVISOR         | ICS'CUTDENOMCELL
        |----------------------------------------|
```

G.23.00
13- 126

Interrupt Control Stack (ICS) Format (Cont.)

```
      0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
     |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
QI-50| TIME IN CPU BEFORE PRIORITY DROP           | ICS'CURCFILTERCELL
     |-----------------------------------------------|
QI-47| MINIMUM TIME IN CPU BEFORE PRIORITY DROP   | ICS'MAXCFILTERCELL*
     |-----------------------------------------------|
QI-46| MAXIMUM TIME IN CPU BEFORE PRIORITY DROP   | ICS'MINCFILTERCELL*
     |-----------------------------------------------|
QI-45| MAXIMUM PRIORITY (LOWEST VALUE)            | ICS'ESCHEDBASECELL*
     | WHEN ON E QUEUE                            |
     |-----------------------------------------------|
QI-44| MAXIMUM PRIORITY (LOWEST VALUE)            | ISC'DSCHEDBASECELL*
     | WHEN ON D QUEUE                            |
     |-----------------------------------------------|
QI-43| MAXIMUM PRIORITY (LOWEST VALUE)            | ICS'CSCHEDBASECELL*
     | WHEN ON C QUEUE                            |
     |-----------------------------------------------|
QI-42| MINIMUM PRIORITY (HIGHEST VALUE)           | ICS'WORSTEPRICELL*
     | WHEN ON E QUEUE                            |
     |-----------------------------------------------|
QI-41| MINIMUM PRIORITY (HIGHEST VALUE)           | ICS'WORSTDPRICELL*
     | WHEN ON D QUEUE                            |
     |-----------------------------------------------|
QI-40| MINIMUM PRIORITY (HIGHEST VALUE)           | ISC'WORSTCPRICELL*
     | WHEN ON C QUEUE                            |
     |--|--|----------------------------------------|
QI-37| |SU|                                       |
     |--|--|----------------------------------------|
QI-36| E QUEUE PRIORITY OSCILLATION ENABLED       |
     |-----------------------------------------------|
QI-35| D QUEUE PRIORITY OSCILLATION ENABLED       |
     |-----------------------------------------------|
QI-34| C QUEUE PRIORITY OSCILLATION ENABLED       |
     |-----------------------------------------------|
QI-33| BOUNDS CHECKING - XDS' BANK ADDRESS        | ISC'XDSEGBNKCELL[64]
     |-----------------------------------------------|
QI-32| BOUNDS CHECKING - XDS' BASE ADDRESS        | ICS'XDSEGBASECELL[64]
     |-----------------------------------------------|
QI-31| BOUNDS CHECKING - LAST VALID XDS' SEGMENT #| ISC'DSEGLIMCELL[64]
     |--|----------------|--------|----------|--------|
QI-30|AM|                |MODE    |          |BNDS| ICS'PMBNDSTATCELL[64]
     |--|----------------|--------|----------|--------|
QI-27|                                           |
     |-----------------------------------------------|
QI-26|                                           |
     |-----------------------------------------------|
QI-25|                PAUSE                       |
     -                TIME                        -
QI-24|               (MPE III ONLY)               |
     |-----------------------------------------------|
```

G.23.00
13- 127

Interrupt Control Stack (ICS) Format (Cont.)

```
      0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
     |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
QI-23| PAUSE CODE (MPE III ONLY)                  |
     |-----------------------------------------------|
QI-22| DISABLE/ENABLE DISPATCHER TO RUN COUNTER   | ICS'PDISCNTCELL**
     |-----------------------------------------------|
QI-21|                RESERVED                    |
     |-----------------------------------------------|
QI-20| CURRENT PROCESS STACK DST NUMBER (FROM PCB)| ICS'STKDSTCELL
     |-----------------------------------------------|
QI-17| PSEUDO INTERRUPT PROCESSOR'S STATUS WORD   | ICS'PISTATUSCELL
     |-----------------------------------------------|
QI-16| BASE ADDRESS OF PSEUDO INTERRUPT PROCESSOR | ICS'PIDELTAPCELL
     |-----------------------------------------------|
QI-15|                                           |
     |-----------------------------------------------|
QI-14|                                           |
     |-----------------------------------------------|
QI-13| ABSOLUTE JOB CUTOFF TABLE ENTRY ADDRESS    | ICS'JCUTCELL
     |-----------------------------------------------|
QI-12| PCB RELATIVE ADDRESS FOR ENTRY OF CUR PROCESS| ICS'CURPCBTCELL
     |-----------------------------------------------|
QI-11| CURRENT PROCESS' BASE ADDRESS TO ITS STACK | ICS'STKBASECELL
     |-----------------------------------------------|
QI-10| CURRENT PROCESS' DB REL VALUE TO Z IN STACK| ICS'STKDBRELZCELL**
     |-----------------------------------------------|
QI- 7| CURRENT PROCESS' DB REL VALUE TO DL IN STACK| ICS'STKDBRELDLCELL**
     |-----------------------------------------------|
QI- 6| CURRENT PROCESS' DB REL VALUE TO S IN STACK| ICS'STKDBRELSCELL**
     |-----------------------------------------------|
QI- 5| CURRENT PROCESS' BANK ADDRESS TO ITS STACK | ICS'STKBANKCELL**
     |-----------------------------------------------|
QI- 4| CURRENT PROCESS' BASE ADDRESS TO DB IN STACK| ICS'ABSSTKDBCELL**
     |-----------------------------------------------|
QI- 3| INITIAL STACK MARKER'S X REGISTER VALUE    |
     |--|--|---------------------------------------|
QI- 2|T |L |  INITIAL STACK MARKER'S P REGISTER VALUE|
     |--|--|---------------------------------------|
QI- 1| INITIAL STACK MARKER'S STATUS WORD         |
     |--|--------------------------------------------|
QI- 0|P |      INITIAL STACK MARKER'S Q VALUE (=0) |
     |--|--------------------------------------------|
QI+ 1| INITIAL STACK MARKER'S DB BANK ADDRESS     |
     |-----------------------------------------------|
QI+ 2| INITIAL STACK MARKER'S DB BASE ADDRESS     |
     |-----------------------------------------------|
QI+ 3| INTERRUPT PARAMETER                        |
     |-----------------------------------------------|
```

*   Tunable by the TUNE command.
**   Known by the firmware.
[64] Series 64 only.

G.23.00
13- 128

QI-45 MEASUREMENT INTERFACE word:
    Bit   0 = In-Motion-In Flag (IM).
            1 = DISC I/O flag bit (IO).
          2-15 = Measurement Interface Word.

QI-31 SIMULATIONS word:
    Bit   1 = 1 if to enable stack underflow simulations call
             by STACKUNDERFLOW in ININ.

QI-24 Privilege Mode Bounds Checking.
    Bit   0 = Absolute Mode - DB and DB bank not matched (AM).

         8-9 = Mode field (MODE).
            = 0 if stack mode - DB = extended CPU register XRA120
               and DB bank = extended CPU register XRB1200.
            = 1 if low core mode - DB<=extended CPU register
               XRA122 and DB bank <=extended CPU register XRB122.
            = 2 if xdseg mode - DB = extended CPU register XRA121
               and DB bank = extended CPU register XRB121.

       14-15 = Bounds check flag (BNDS).

            = 0 if DB, Q, and S bounds enabled.
            = 1 if DB bounds disabled, Q and S bounds disabled.
            = 2 DB bounds enabled, Q and S bounds disabled.
            = 3 DB, Q, and S bounds enabled.

QI- 2 Initial stack marker's P word.
    Bit   0 = TRACE enabled flag bit (T).
            1 = logically/physically mapped code segment (L).
         2-15 = program location value.

QI- 0 Initial stack marker's Q word.
    Bit   0 = 1 if there is a pending DSP that cannot be
             processed immediately (e.g., DISPATCHER
             was PDISABLED or on the ICS).
        1-15 = 0 (indicating no previous stack marker).

QI+ 3 Interrupt Parameter word.
      <1 if External Program Label parameter.
      >0 if a parameter that is passed to internal interrupt
      handler.

G.23.00
13- 129

ICS Global Cells With Initial Values

ICS'ABSSTKDBCELL - Absolute address of the currently running
               process' stack.

ICS'STKBANKCELL   - Bank address for process' stack.

ICS'STKDBRELSCELL - Stack DB relative S.

ICS'STKDBRELDLCELL - Stack DB relative DL.

ICS'STKDBRELZCELL - Stack DB relative Z.

ICS'STKBASECELL   - Absolute stack address.

ICS'CURPCBTCELL   - PCB table relative pointer to word 0 of the
               running process' Process Control Block.

The above cells are to be initialized for the PROGENITOR.

ICS'STKDSTCELL   - DST number for running process' stack.

ICS'JCUTCELL   - The bank 0 absolute address of the JCUT
               (Job Cutoff) Table.

ICS'PIDELTAPCELL - PB relative address for the procedure
               PSEUDOINT (handles pseudo/soft interrupts)

ICS'PISTATUSCELL - Status value for PSEUDOINT (X40000+CST#)

ICS'PDISCNTCELL   - PSDB counter, initially 0

INITIAL sets the above as described.

G.23.00
13- 130

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                      |ICPVR0
  1|          PROGRAM                      |ICPVR1
  2|          VARIABLE                     |ICPVR2
  3|              AREA (ICPVR)             |ICPVR3
   |---------------------------------------|
  4|     DMA ABORT                         |ICPVR4
  5|     ADDRESS                           |ICPVR5
   |---------------------------------------|
  6|              0                        |ISRQL
   |--|------------------|-----|-------|---|
  7|LIM| 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                         |
   |---------------------------------------|
```

ICPVR0 - 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.

ICPVR4 - 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 the channel code when this program halts.

HCUNIT -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 (X55)   ODD: DST = 46 (X56)
         SIR = 3   SIR = 4

Overview of Table Structure

```
   |-----------------------------------|
  0| ENTRY 0 (8 WORDS)                 |
   |-----------------------------------|
  2| SUBENTRY AREA POINTER             |---\
   |-----------------------------------|   |
  7|                                   |   |
   |-----------------------------------|   |
   |                                   |   |
   | HEAD ENTRIES (4 WORDS EACH)       |   |
   |                                   |   |
   |-----------------------------------|   |
   |                                   |<--/
   | SUBENTRIES (X40 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 = X40|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 out-fence. 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 entries; a class entry and a logical device entry. There is only one class entry; it is the first head entry in the ODD. The IDD does not have a class entry; position is filled with zeros. All spoolfiles opened by class (e.g., LP, SLOWLP, EPOC, PP) 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 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    |CLI|                        |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|IO| 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              |
|    |----------------------------------------|
|  30| NUMBER OF EXTENTS   |                  |24
|    |----------------------------------------|
|  31| LAST EXTENT SIZE (SECTORS)             |25
|    |--|--|--|--|--|--|---------------------|
|  32|SQ|  |RS|FD|SO|A9|   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
                  requests 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.
                 Simulates 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
                 device 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
                 acquire a new extent when creating spoolfile.
        AB   - This is the $STDLIST of an aborted job.
Words %36-37   - Time stamp when spoolfile 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)         |
             |-------------------------------------------|
Word 24:     | DATE (FROM CALENDAR INTRINSIC)             |
             |-------------------------------------------|
Words 25&26: | TIME (FROM CLOCK INTRINSIC)                |
             |-------------------------------------------|
Words 30&31: | "MPEV" 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 1020 word records as needed. The last record will be padded with zeros if necessary. The entry format is:

```
             0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
           |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
Word 0:  |O| DEVICE FILE ID NUMBER                         |
           |--|--------------------------------------------|
Words 1-3:|                    0                           |
           |-----------------------------------------------|
Words 4-7:|  USER NAME                                     |
           |-----------------------------------------------|
Words 8-11:| ACCOUNT NAME                                  |
           |-----------------------------------------------|

         O = 1  File is an output spoolfile
```

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

```
          0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
        |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
Word 0: |  LOGICAL DEVICE NUMBER                         |
        |-------------------------|----------------------|
Word 1: | DEVICE SUBTYPE          | LENGTH OF ENTRY (3)  |
        |-------------------------|----------------------|
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|
| SPOOLFILE BLOCK        INTO ONE 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 under the User Labels Information section below.

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            |
           |          X100 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 five tape records actually contain user labels in addition to the FOPEN records. The user labels are packed three to a spoolfile block, six to a tape record. Each spoolfile block of 512 words has the following format:

```
           |-------------------------------------------|
Words 0-4: | FOPEN RECORD                              |
           |-------------------------------------------|
           | TO TERMINATE THE BLOCK                    |
           |-------------------------------------------|
Words X200-X377:| USER LABEL                           |
           |-------------------------------------------|
Words X400-X577:| USER LABEL                           |
           |-------------------------------------------|
Words X600-X777:| 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  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
    |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
  0 | NUMBER OF ENTRIES                              |   \
    |-----------------------------------------------|   |
  1 | MAX NUMBER OF ENTRIES                          |   |
    |-----------------------------------------------|   |
  2 | POSITION OF NEXT FREE ENTRY SPACE IN QUEUE     |   | TABLE
    |-----------------------------------------------|   | HEADER
  3 | NUMBER OF QUEUED ENTRIES                       |   | (57 words)
    |-----------------------------------------------|   |
    | (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 | DB Offset                                      |   |
    |-----------------------------------------------|   |
  5 | Ldev                                           |   |
    |-----------------------------------------------|   |
  6 |                                                |   | ENTRY
    |-----------------------------------------------|   | (51 words)
  7 |           # BYTES IN MESSAGE                   |   |
    |-----------------------------------------------|   |
    |                                                |   |
    | MESSAGE IN ASCII                               |   |
    |                                                |   |
    | (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 number
             = 4 for string
   TABLE SIZE = 2046 words
   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 user label).

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 with 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 format designator INPUT, which must be used for all entry points. The program has the following entry points:

(no entry - Reads from input file and builds a temporary file
point)   (formal designator CATALOG). Also renames any old temporary CATALOG, CATnn, using an archival numbering scheme (i.e., CAT1, CAT2).

BUILD - (Must log on under MANAGER.SYS.) Reads from input file, build the system message catalog (formal designator 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 user label 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=mm. 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 (formal designator 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
$SET 28 - Ciper Device messages
$SET 29 - Store/Restore messages

Message Set Directory

- DST # in SYSGLOB X373
- CAT DISC ADDR in SYSGLOB X371-372
- 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 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
       |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
     0| DIRECTORY SIZE (WORDS)                       |
       |--------------------------------------------|
     1| ENTRY LENGTH (BYTES) | KEYWORD LENGTH (BYTES)| \
       |----------------------|---------------------| |
     2| ENTRY                                        | |
     | KEYWORD                                       | | ENTRY
     ~ 1-255 BYTES                                   ~ |
     |                                               | |
       |--------------------------------------------| |
       |      ENTRY RECORD # IN CICAT                | |
       | LEFT BYTE           | RIGHT BYTE            | /
       |---------------------|-----------------------|
       | ENTRY LENGTH (BYTES)| KEYWORD LENGTH (BYTES)| \
       |---------------------|-----------------------| |
     | ENTRY                                          | |
     | KEYWORD                                        | | ENTRY
     ~ 1-255 BYTES                                    ~ |
     |                                                | |
       |---------------------|-----------------------| |
       |ENTRY REC # RIGHT BYTE| ENTRY REC # LEFT BYTE | /
       |----------------------|----------------------|
       |KEYWORD LENGTH (BYTES)| ENTRY LENGTH (BYTES)  | \
       |----------------------|----------------------| |
     | ENTRY                                          | |
     | KEYWORD                                        | | ENTRY
     ~ 1-255 BYTES                                    ~ |
     |                                                | |
       |----------------------|----------------------| |
       | ENTRY REC #          |                      | |
       | LEFT BYTE            | RIGHT BYTE            | /
       |----------------------|----------------------|
     ~                                                ~
     |                                                |
       |--------------------------------------------|
```

UDC Directory

- Extra Data Segment - DST # in DB+X255 of URRIN Stack
- Built by INITUDC

```
        0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
       |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
       |LT|LM|NH|NB|      | TYPE|  ENTRY SIZE       | \
       |--|--|--|--|------|-----|-------------------| |
       | HEADER RECORD NUMBER                        | |
       |--------------------------------------------| |
       | BODY RECORD NUMBER                          | |
       |--------------------------------------------| |
       | FILE NUMBER          | COMMAND LENGTH       | |
       |----------------------|----------------------| >
     | COMMAND                                        | |
     | NAME                                           | |
     ~ (1-16 BYTES)                                   ~ |
     |                                                | |
       |--------------------------------------------| /
     ~                                                ~
     | ENTRIES                                        |
     |                                                |
       |--------------------------------------------|
     | LAST COMMAND ENTRY                             |
     |                                                |
       |--------------------------------------------|
     | LAST ENTRY (12 WORDS OF ZEROS (0))             |
     ~                                                ~
     |                                                |
       |----------------------|----------------------|
       |                      |        0             |  ENTRY SIZE=0
       |----------------------|----------------------|  ENDS DIRECTORY

      LT = OPTION LIST
      LM = OPTION LOGON
      NH = OPTION NOHELP
      NB = OPTION NOBREAK
      TYPE = 00 - USER UDC
             01 - ACCOUNT UDC
             10 - SYSTEM UDC
```

UDCs 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 UDCs
- Can be rebuilt to re-enable UDCs

```
   X   RECORD 0    #      X    FREE ENTRY    #
   |----------------|     |-------------------|
 0 |1ST FREE ENTRY #|0   0|NEXT FREE ENTRY #|0
   |----------------|     |-------------------|
 1 |    NOT USED    |1   1|  ENTRY TYPE=0   |1
   |----------------|2    |-----------------|
 2 |   MAX IN USE   |2   2|                 |2
   |----------------|3    |                 |
 3 |   # IN USE     |3    -    NOT USED      -
   |----------------|4    |                 |
 4 |                |4    |                 |
   |   NOT USED     |     |                 |
   -                -     -                 -
   |                |     |                 |
23 |                |19 23|                 |19
   |----------------|     |-----------------|
   X   USER ENTRY   #      X   FILE ENTRY    #
   |----------------|     |-----------------|
 0 | CATALOG ENTRY #|0   0|NEXT CAT. ENTRY #|0
   |----------------|     |-----------------|
 1 |  ENTRY TYPE=1  |1   1|  ENTRY TYPE=2   |1
   |----------------|     |-----------------|
```

UDCs COMMAND.PUB.SYS (Cont.)

```
   |----------------|      |-----------------|
 2 |                |2   2 |   FILE NAME     |2
 3 |    USER*       |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 UDCs.

  If only the User Field contains @ and 7 spaces, this indicates
  Account Level UDCs.

CI Stack Definition

```
          0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
          |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
  DB+X0 | BCOMIMAGE (BYTE PTR. TO COMMAND)           |
          |-------------------------------------------|
  DB+X1 | COMMAND IMAGE                              |
        | (280 BYTES)                                |
        -                                            -
          |-------------------------------------------|
  DB+X215 | LINELENSTACK                             |
          | (30 WORDS)                               |
        -                                            -
          |-------------------------------------------|
  DB+X253 | NEXTMSG (NOT CURRENTLY USED)             |
          |-------------------------------------------|
  DB+X254 | (NOT USED)                               |
          |-------------------------------------------|
  DB+X255 |                                          | UDC0
          |-------------------------------------------|
  DB+X256 |                                          | UDC1
          |--|----------------------------------------|
  DB+X257 |FL|                                        | UDC2
          |--|--|--|--|-------------------------------|
  DB+X260 |LI|LM|NH|NB|                               | UDC3
          |--|--|--|--|--|----------------------------|
  DB+X261 |FE|EB|BK|NP|IA|        NESTLEVEL          | UDC4
          |--|--|--|--|--|----------------------------|
  DB+X262 | IFNESTING                                |
          |-------------------------------------------|
  DB+X263 | IFSKIP                                   |
          |-------------------------------------------|
  DB+X264 | ELSESEEN                                 |
          |-------------------------------|--|--|--|
  DB+X265 |                               |SQ|  |WR| CIFLAGS
          |-------------------------------|--|--|--|
  DB+X266 |          CONTINUE            |
          |---        STATE              ---|
  DB+X267 |           STACK              |
          |-------------------------------------------|
  DB+X270 | PENDINGCOMLEN                            |
          |-------------------------------------------|
  DB+X271 | BLASTCOMIMAGE (BYTE PTR.)                |
          |-------------------------------------------|
  DB+X272 | 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.

UDC0   -   Holds the DST number of the UDC definitions.

UDC1   -   Holds the old S register value for UDCs.

UDC2: (0:1) -   FLUSHUDC, used by :SETCATALOG

UDC3: (0:1) -   OPTION LIST = 1
      (1:1) -   OPTION LOGON = 1
      (2:1) -   OPTION NOHELP = 1
      (3:1) -   OPTION NOBREAK = 1

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|                                    |0      DST X42
|  1|                                    |1
|  2| NOT                                |2      SIR X30
|  3|                                    |3
|  4| USED                               |4
|  5|                                    |5      ONE ENTRY/
|  6|                                    |6      SYSTEM LDEV
|----------------------------------------|
|  7| JMAT INDEX                         |7    \
|----------------------------------------|
| 10| JIT DST NUMBER                     |8    |
|----------------------------------------|
| 11| DST REL. INDEX TO USER'S NEXT ENTRY|9    - Ldev 1
|----------------------------------------|
| 12| CLASS NAME UNDER WHICH THIS LDEV IS|10   |  (ASSOCIATED)
| 13| ASSOCIATED> LEFT JUSTIFIED AND     |11   |
| 14| PADDED WITH BLANKS.  8 BYTES.      |12   |
| 15|                                    |13   /
|----------------------------------------|
| 16|            0                       |14   \
|----------------------------------------|
| 17|            0                       |15   |
|----------------------------------------|
| 20|            0                       |16   - LDEV 2
|----------------------------------------|
| 21|                                    |17   |  (UNASSOCIATED)
| 22| UNDEFINED                          |18   |
| 23|                                    |19   |
| 24|                                    |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.           |    /
|----------------------------------------|
```

Application Message Facility

The Application Message Facility consists of two parts: GENCAT, the catalog maintenance facility, and the "CAT" intrinsics, through which the message catalogs are accessed. The "compiled" catalog, which GENCAT creates, contains an extensive directory at the front of the file which describes where every message in the catalog is located. When a message catalog is opened (via CATOPEN) part of this directory is read into an extra data segment which is created specifically for that purpose. This "caching" of the directory provides nearly direct access to the desired message.

These messages include message set number, message numbers, and record numbers placed or "cached" into 384 word message caches. The first set number and message number of each message cache is placed into a cache directory (set and message numbers must be ascending). A message is found by scanning first the cache directory, then the message cache searching for the desired set and message number. The retrieved message directory entry contains the record number in the catalog file of that message. Now, the catalog file can be read directly using the record number.

Internally, the two layer directory format is used by both the formatted application message catalog, and the message extra data segment created by the intrinsic CATOPEN (and used by CATREAD).

The catalog files created for MAKECAT and GENCAT may be used with the Application Message Facility. In most cases, applications will increase their performance in message routing and decrease the file space with formatted catalogs.

NLS Message Catalog/DST Overview

```
                    |-----------|
                    | CACHE DIRECTORY |
                    |-----------|
        |------------------|  |  |------------------|
        |      |-----V----|  |  |-----V----|      |
    |----V---| | MESSAGE DIRECTORIES |  |---V----|
    |--------| |----------|  |----------|  |--------|
   |---| | |---|   |---| | |---|   |---| | |---|
  |--V--|--V--|--V-----V----V-----V----V-----V----V---|
  |DATA | DATA | DATA| DATA | DATA | DATA | DATA | DATA | DATA |
  |------------------------------------------------------------|
```
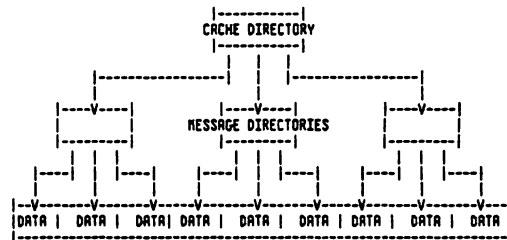
The maximum catalog size is 65536 sectors long. The largest set number is 255. The largest message number is 64766, while the smallest set and message number is 1.

Formatted Catalog File Structure

```
       |-------------------------|
REC# 0 |   OVERHEAD              |
REC# 1 |-------------------------|
       |- CACHE DIRECTORY      - |
       |                         |
REC# MC|-------------------------|
       |                         |
       |- MESSAGE CACHES       - |
       |                         |
REC# D |-------------------------|
       |                         |
       |- DATA                 - |
       |                         |
       |-------------------------|
```

Where: MC = 2 + (2 * #message caches)/128
        D = MC + (384 * #message caches)/128

Each physical record is one sector long (128 words). Each structure starts on a sector boundary.

Cache Directory

Each entry in the cache directory is a two-word entry. There exists one cache directory entry for each 384-word message cache. The first word of the cache directory entry is the set number of the first entry in the associated message cache. The second word of the cache directory entry is the message number of the first entry in the associated message cache.

```
       |---------------------------|
       | 1ST SET NUMBER OF CACHE 1 |<-|
       |---------------------------|  |-CACHE DIRECTORY ENTRY 1
       | 1ST MSG NUMBER OF CACHE 1 |<-|
       |---------------------------|
       | 1ST SET NUMBER OF CACHE 2 |<-|
       |---------------------------|  |-CACHE DIRECTORY ENTRY 2
       | 1ST MSG NUMBER OF CACHE 2 |<-|
       |---------------------------|
       |-                         -|
       |                           |
       |---------------------------|
       | 1ST SET NUMBER OF CACHE N |<-|
       |---------------------------|  |-CACHE DIRECTORY ENTRY N
       | 1ST MSG NUMBER OF CACHE N |<-|
       |---------------------------|
```

Message Cache Format

Each message cache is 384 words long (3 records). A message cache entry is 3 words long, 128 entries per message cache. Each entry contains the message number and set number of the message. The byte offset is the offset to the start of the message in the record specified by the record number. Entry 127 is a duplicate of the first entry in the next cache. This is to allow the total number of bytes of the message to be computed without reading the next message cache.

```
|-----------------------------|
|       MESSAGE NUMBER        |<-      <-----|
|-----------------------------|      |        |
|  SET NUMBER  | BYTE OFFSET  |      |-ENTRY 0|
|-----------------------------|      |        |
|       RECORD NUMBER         |<-    |
|-----------------------------|      |
|                             |      |
|-                          - |      |-CACHE 1
|                             |
|-----------------------------|
|       MESSAGE NUMBER        |<-
|-----------------------------|      |
|  SET NUMBER  | BYTE OFFSET  |      |-ENTRY 128
|-----------------------------|      |
|       RECORD NUMBER         |<-      <-----|
|-----------------------------|
|                             |
|-                          - |
|-----------------------------|
|       MESSAGE NUMBER        |<-      <-----|
|-----------------------------|      |        |
|  SET NUMBER  | BYTE OFFSET  |      |-ENTRY 0|
|-----------------------------|      |        |
|       RECORD NUMBER         |<-    |
|-----------------------------|
|                             |
|-                          - |      |-CACHE N
|                             |
|-----------------------------|
|       MESSAGE NUMBER        |<-
|-----------------------------|      |
|  SET NUMBER  | BYTE OFFSET  |      |-ENTRY 127
|-----------------------------|      |
|       RECORD NUMBER         |<-      <-----|
|-----------------------------|
```

Data Format

The format of the messages is straightforward. It contains only the text of the message. It contains no comment records, message numbers or set numbers. All leading and trailing blanks are stripped from the message.

Message DST (MDST) Structure

A message extra data segment is allocated during a CATOPEN. The data segment number is kept by the application on the return from CATOPEN. The format of the data segment is similar of that of the formatted message catalog. The main difference is the addition of a table to track resident caches in the DST, and the catalog data is not kept in the DST.

Message DST Overview

```
|-----------------------------|
|                             |MDST'OVERHEAD
|    MDST OVERHEAD            |
|                             |
|-----------------------------|
|                             |MDST'RESIDENT'CACHE
|- MDST RESIDENT CACHE AREA  -|
|                             |
|-----------------------------|
|                             |MDST'CACHE'DIR
|- MDST CACHE DIRECTORY      -|
|                             |
|-----------------------------|
|                             |MDST'MSG'DIR
|    MDST RESIDENT MESSAGE    |
|-                           -|
|    CACHES                   |
|-----------------------------|
```

NOTE: A resident cache is a message cache copied from the formatted catalog. Resident caches are swapped in and out of the MDST and are used to determine the record number of the desired set and message.

Message DST Overhead

```
     |-------------|-------------|
   0 |    "M"      |    "D"      |MDST'ID
     |-------------|-------------|
   1 |    "S"      |    "T"      |
     |-------------|-------------|
   2 |  SIZE OF MDST ( IN WORDS )|MDST'SIZE
     |---------------------------|
   3 |  CATALOG FILE NUMBER      |MDST'CAT'FNUM
     |---------------------------|
   4 |  OFFSET TO RESIDENT CACHE |MDST'RESIDENT'CACHE
     |---------------------------|
   5 |  OFFSET TO CACHE DIRECTORY|MDST'CACHE'DIR
     |---------------------------|
   6 |  OFFSET TO MSG DIRECTORIES|MDST'MSG'DIR
     |---------------------------|
   7 |  CACHE DIRECTORY SIZE (WDS)|MDST'CDIR'SIZE
     |---------------------------|
  10 |  MSG DIRECTORY SIZE (WDS) |MDST'DIR'SIZE
     |---------------------------|
  11 |  MAX NUM OF RESIDENT CACHE |MDST'CACHE'MAX
     |---------------------------|
  12 |  RECNUM OF FIRST MSG DIR. |MDST'FIRSTDIR'RECNUM
     |---------------------------|
  13 |  RESERVED                 |
     |---------------------------|
  14 |  RESERVED                 |
     |---------------------------|
```

Message DST Resident Cache Area

The Resident Cache Area is a table of the message directory blocks currently stored in the MDST, together with their index. They are held in order from the most recently accessed at the top and the oldest on the bottom. The maximum number of caches held in the MDST at any one time is MDST'CACHE'MAX.

```
|-------------------------------|
| CACHE W MOST RECENT ACCESS |MDST'RESIDENT'CACHE
|-------------------------------|
| INDEX OF CACHE DIRECTORY      |
| MOST RECENT ACCESS CACHE      |
|-------------------------------|
|                               |
|-                             -|
|                               |
|-------------------------------|
```

MDST Cache Directory

Each entry in the cache directory is a two-word entry. There exists one cache directory entry for each 384 word message cache. The first word of the cache directory entry is the set number of the first entry in the associated message cache. The second word of the cache directory entry is the message number of the first entry in the associated message cache.

```
|-----------------------------|
| 1ST SET NUMBER OF CACHE 1 |<-|
|-----------------------------|  |-CACHE DIRECTORY ENTRY 1
| 1ST MSG NUMBER OF CACHE 1 |<-|
|-----------------------------|
| 1ST SET NUMBER OF CACHE 2 |<-|
|-----------------------------|  |-CACHE DIRECTORY ENTRY 2
| 1ST MSG NUMBER OF CACHE 2 |<-|
|-----------------------------|
|                             |
|-                           -|
|-----------------------------|
| 1ST SET NUMBER OF CACHE N |<-|
|-----------------------------|  |-CACHE DIRECTORY ENTRY N
| 1ST MSG NUMBER OF CACHE N |<-|
|-----------------------------|
```

MDST Message Cache Format

Each message cache is 384 words long (3 records). A message cache entry is 3 words long, 128 entries per message cache. Each entry contains the message number and set number of the message. The byte offset is the offset to the start of the message in the record specified by the record number. Entry 127 is a duplicate of the first entry in the next cache. This is to allow the total number of bytes of the message to be computed without reading the next message cache.

```
          |-----------------------|
          |    MESSAGE NUMBER     |<-    <-----|
          |-----------------------|            |
          | SET NUMBER|BYTE OFFSET|  |-ENTRY 0  |
          |-----------------------|            |
          |    RECORD NUMBER      |<-          |
          |-----------------------|            |
          |                       |            |
          -                       -  |-CACHE 1
          |                       |            |
          |-----------------------|            |
          |    MESSAGE NUMBER     |<-          |
          |-----------------------|            |
          | SET NUMBER|BYTE OFFSET|  |-ENTRY 128|
          |-----------------------|            |
          |    RECORD NUMBER      |<-   <-----|
          |-----------------------|
          |                       |
          -                       -
          |                       |
          |-----------------------|
          |    MESSAGE NUMBER     |<-   <-----|
          |-----------------------|            |
          | SET NUMBER|BYTE OFFSET|  |-ENTRY 0  |
          |-----------------------|            |
          |    RECORD NUMBER      |<-          |
          |-----------------------|            |
          |                       |            |
          -                       -  |-CACHE N
          |                       |            |
          |-----------------------|            |
          |    MESSAGE NUMBER     |<-          |
          |-----------------------|            |
          | SET NUMBER|BYTE OFFSET|  |-ENTRY 127|
          |-----------------------|            |
          |    RECORD NUMBER      |<-   <-----|
          |-----------------------|
```

CHAPTER 16 SYSDUMP/INITIAL

CONFDATA File

Record 0 of CONFDATA File (CTAB0)

```
   0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
 |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
0| CHECKSUM OF CTAB (REC 0)            |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 LOGON              |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| MAXIMUM # OF SPOOL FILES (KILO SECTORS) |30
37|                                   |31
40|                                   |32
41| # SECTORS PER SPOOL EXTENT        |33
```

Record 1 of CONFDATA File (CTAB)

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

Record 1 of CONFDATA File (CTAB) (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
```

INITIAL/PROGEN Communication DST

The INITIAL/PROGEN Communication data segment is used by Initial to pass information to PROGEN. This segment is only temporary and not memory resident.

COMMDSTN = SYSGLOBEXT (%122)   DST (SYSGLOBEXT (%122))

```
             0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 #
             |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
/----|  POINTER TO THE START OF CTABO           |0
|    |----------------------------------------------|
| /----|  POINTER TO THE START OF CTAB          |1
| |  |----------------------------------------------|
| |  |  SYSTEM START-UP OPTION                  |2   OPT
| |  |----------------------------------------------|
| |  |  RECOVER LOST DISC SPACE PROGRAM         |3   Recovery
| |  |----------------------------------------------|
| |  |  RESERVED                                |
| |  |----------------------------------------------|
\---> |  CTABO ARRAY (RECORD 0 OF THE CONFDATA FILE) |256 = %400
  |  |----------------------------------------------|
  \--->|  CTAB ARRAY (RECORD 1 OF THE CONFDATA FILE) |256 + CTABO size
     |----------------------------------------------|
```

DESCRIPTIONS

```
OPT = Start-up option        Recovery = 1 If Recover Lost Disc Space
      0 = Warmstart
      1 = Coolstart                   = 0 If Not Recover Lost Disc Space
      2 = Coldstart
      3 = Update
      4 = Reload         CTAB & CTABO - See the descriptions of CONFDATA
                                        file in this chapter.
```

The microcode will store the CNTRL B command into (QI-11) equivalent to (ABS(5)-11) for the Series 37.

```
CNTRL B  0 = Start
         1 = Warmstart
         2 = Coolstart
        %10 = Load
        %11 = Update
        %12 = Coldstart
        %13 = Reload
        %14 = New
        %20 = Dump
```

Starttype = ABS (ABS (5)-11)

DEFDATA Table Lookup File

This file contains the default information for HP-supported devices. This file, DEFDATA.PUB.SYS, is available to Sysdump and Initial and eliminates the necessity for looking up default information every time a device is added to the system. Despite its name, DEFDATA.PUB.SYS in not only a file, but a table in the Coldload Information Table. It is not easily modified. Therefore, it is recommended that the file be left alone; if any user is unhappy with the defaults, they can be overridden during the Sysdump or Initial dialogues.

DEFDATA Table Lookup File Header Format

```
     0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
     |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
  0| CHECKSUM                                       |
     |------------------------------------------------|
  1| VERSION                                        |
     |------------------------------------------------|
  2| TOTAL TABLE SIZE IN WORDS                      |
     |------------------------------------------------|
  3| ENTRY SIZE (SET TO 1)                          |
     |------------------------------------------------|
  4| # OF TABLE ENTRIES                             |
     |------------------------------------------------|
```

DEFDATA Table Lookup File Entry Format

```
     0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
     |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
  0|                                                |
  1|                                                |
  2|  DEVICE  NAME                                  |
  3|                                                |
  4|                                                |
  5|                                                |
  6|                                                |
  7|                                                |
     |------------------------------------------------|
 10| TOTAL DEVICE ENTRY SIZE (IN WORDS)             |
     |------------------------------------------------|
 11| # OF DEVICE CLASSES FOR THIS DEVICE (SET TO 1) |
     |------------------------------------------------|
```

DEFDATA Table Lookup File Entry Format (Cont.)

```
           0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
           |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
/---12|DEVICE CLASS NAME LIST POINTER (ENTRY RELATIVE)|
|   |---------------------------------------------------|
|/--13|TERMINAL DESCR. FILE NAME POINTER (ENTRY REL.)  |
||  |---------------------------------------------------|
|||/-14|  DEFAULT OUTPUT DEV. OR POINTER TO DEVCLASS   | (ENTRY RELATIVE)
|||| |----------------------------------------------------|
|||| 15|  CS LDTX ENTRY POINTER (CURRENTLY SET TO 0)  |
|||| |----------------------------------------------------|
|||| 16|  RESERVED                                    |
|||| |----------------------------------------------------|
|||| 17|  DEVICE ID CODE                              |
|||| |----------------------------------------------------|
|||| 20|  RESERVED                                    |
|||| |----------------------------------------------------|
|||| 21|  RESERVED                                    |
|||| |------------------|--|--|--|--|--|-----------------|
|||| 22|  DEVICE TYPE    | SUBTYPE |J | A| I| D| SS |      J=Job Accepting
||||   |                 |         |  |  |  |  |    |      A=Data Accepting
||||   |                 |         |  |  |  |  |    |      I=Interactive
||||   |                 |         |  |  |  |  |    |      D=Duplicative
||||   |                 |         |  |  |  |  |    |      SS=Spool State
|||| 23|  CHAN. #|CR|DS|SQ|CL|AI| RECORD WIDTH |      CR=Core Resident
||||   |                                            DS=DS Device
||||   |                                            SQ=Spool Queues
||||   |                                            CL=Indicates
||||   |                                               whether the
||||   |                                               output device
||||   |                                               is given.
||||   |                                            AI=Default Auto
||||   |                                               Increment (DRT
||||   |                                               or Unit)
|||| |------------------|--|--------------------|
|||| 24| DEFAULT TERM. TYPE |AR| RESERVED       |      AR=Auto Reply
|||| |------------------|--|--------------------|
|||| 25| TERM SPEED                             |
|||| |----------------------------------------------------|
|||| 26| RESERVED                               |
|||| |----------------------------------------------------|
|||| 27| RESERVED                               |
|||| |----------------------------------------------------|
|||| 30|                                        |
|||| 31|                                        |
||||   | DRIVER NAME                            |
|||| 32|                                        |
```

DEFDATA Table Lookup File Entry Format (Cont.)

```
|||  33|-                                       -|
|||    |------------------------------------------|
|\->34|- TERMINAL DESCRIPTOR FILE NAME          -|
| |    |-                                       -|
| |    |-                                       -|
| |    |- TERMINAL DESCRIPTOR GROUP NAME        -|
| |    |-                                       -|
| |    |- TERMINAL DESCRIPTOR ACCOUNT NAME      -|
| |    |-                                       -|
| |    |------------------------------------------|
|\-->|-                                         -|
|    |- OUTPUT DEVICE CLASS NAME                -|
|    |-                                         -|
|    |------------------------------------------|
\---->|-                                        -|
     |- DEVICE CLASS NAME                       -|
     |-                                         -|
     |------------------------------------------|
     |  RESERVED                                |
     |------------------------------------------|
```

DEVDATA.PUB.SYS

Overview

```
|---------------------------|
|  PARAMETER RECORD         |
|---------------------------|
|  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        |
    |---------------------------|
```

Parameter Record (Cont.)

```
|    %100|  REC #     |-------| DVR TABLE
|        |--          |--     |
|        |  LENGTH    |-------|
|    %102|  REC #     |-------| LPDT
|        |--          |--     |
|        |  LENGTH    |-------|
|    %104|  REC #     |-------| LDT
|        |--          |--     |
|        |  LENGTH    |-------|
|    %106|  REC #     |-------| LDTX
|        |--          |--     |
|        |  LENGTH    |-------|
|    %110|  REC #     |-------| DCTH
|        |--          |--     |
|        |  LENGTH    |-------|
|    %112|  REC #     |-------| CLASS
|        |--          |--     |
|        |  LENGTH    |-------|
|    %114|  REC #     |-------| TERM DEF
|        |--          |--     |
|        |  LENGTH    |-------|
|    %116|  REC #     |-------| ADD'L DVR
|        |--          |--     |
|        |  LENGTH    |-------|
|    %120|  REC #     |-------| CS DEF
|        |--          |--     |
|        |  LENGTH    |-------|
|    %122|  REC #     |-------| CS TABLE
|        |--          |--     |
|        |  LENGTH    |-------|
|        |-           -|
|    %200|- UNUSED     -|
|        |-------------|
```

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
  |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
 0| DRT #                                     |
  |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
 1|CR| CHAN #       |DS|  UNIT #              |
  |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
 2| MASTER LDEV                               |  TYPICAL ENTRY
  |-------------------------------------------|  FORMAT
 3|          D          |         R           |
  |-------------------------------------------|
 4|          I          |         V           |
  |-------------------------------------------|
 5|          N          |         A           |
  |-------------------------------------------|
 6|          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       |
--| UCS TABLE PRT               |95
  |------------------------------|127
 || AMIGO                      |
 ||------------------------------|
->| UCS TABLE                   |
  |------------------------------|
 | UCS #1                       |
  |------------------------------|
 | CHECKSUM      *              |  * Appear only if
 | AMIGO                        |  SYSUCS64 is
  |------------------------------|  present.  Skips
 | AMIGO         *              |  to next CHECKSUM
  |------------------------------|  AMIGO.
```

SYSDUMP Format (Cont.)

```
|------------------------------|
|  UCS #2                      |  Only for the 64/68.  Refer to the
|------------------------------|  UCS Table for the 64/68 below.
|  UCS #N                      |
|------------------------------|<---ENTRY POINT #2 (UCS BASED
|  CHECKSUM                    |0   MACHINES)
|  AMIGO                       |
|------------------------------|127
|  AMIGO                       |
|------------------------------|
|  ICS                         |
|------------------------------|
|  LOW CORE                    |
|------------------------------|
|  INITIAL CST                 |
|------------------------------|
|  CS TABLE                    |
|------------------------------|
|  DEVICE CLASS TABLE HEADER   |
|------------------------------|
|  DEVICE CLASS TABLE          |
|------------------------------|
|  TERMINAL DESCRIPTOR TABLE   |
|------------------------------|
|  TABLE LOOKUP BUFFER         |
|------------------------------|
|  VTAB                        |
|------------------------------|
|  OLDVTAB                     |  *
|------------------------------|
|  DISC COLD LOAD INFORMATION TABLE |  *
|------------------------------|
|  CTAB                        |
|------------------------------|
|  CTAB0                       |
|------------------------------|
|  COMMUNICATION RECORD        |
|------------------------------|
|  CSDVR                       |
|------------------------------|
|  CSDEF                       |
|------------------------------|
```

SYSDUMP Format (Cont.)

```
|-------------------------------------|
| INITIAL'S DB AREA                   |
|-------------------------------------|
| STACK MARKER                        |
|-------------------------------------|
| DRIVER TABLE                        |
|-------------------------------------|
| LPDT                                |
|-------------------------------------|
| LDT                                 |
|-------------------------------------|
| LDTX                                |
|-------------------------------------|
| INITIAL'S SEGMENTS                  |
|-------------------------------------|
| RIN TABLE                           | *
|-------------------------------------|
| LOGGING IDENTIFIER TABLE            | *
|-------------------------------------|
| DIRECTORY HEADER                    | *
|-------------------------------------|
| DIRECTORY                           | *
|-------------------------------------|
| EOF                                 |
|-------------------------------------|
| SYSTEM PROGRAMS, SL, NON-STD. DRIVERS |
|-------------------------------------|
| EOF                                 |
|-------------------------------------|
| STORE/RESTORE HEADER                |
|-------------------------------------|
| EOF                                 |
|-------------------------------------|
| STORE/RESTORE DIRECTORY             | *
|-------------------------------------|
| EOF                                 |
|-------------------------------------|
| USER FILES (SEPARATED BY "EOF'S"    | *
|-------------------------------------|
| STORE/RESTORE TRAILER               |
|-------------------------------------|
| EOF                                 |
|-------------------------------------|
| EOF                                 |
|-------------------------------------|
| EOF                                 |
|-------------------------------------|
```

* NOT DUMPED IF DATE = CARRIAGE RETURN

Note: On disc, READ-SIO-PROGRAM kept in Disc Label.

WCS Table Format

```
|-------------------------------------|
| 0| # RECORDS TO WCS *              |0
|   | # RECORDS OF WCS               |
|   | # RECORDS AFTER WCS            |
|   | WCS RECORD SIZE ON TAPE        |
|-------------------------------------|
| 1|                                 |1
|-------------------------------------|
| 2|                                 |2
|-------------------------------------|
| 3|                                 |3
|-------------------------------------|
| 4|                                 |4
|-------------------------------------|
|   |                                |
|-------------------------------------|
| X45|                               |37
|-------------------------------------|
```

* If SYSWCS64 is present, #WCS records following =0.
  If SYSWCS64 is not present, the preceding entry is repeated.

Series 6x/70 WCS Table Format

One entry (Entry 4) is used by Series 64, 68, and 70.

```
|---------|-----------|---------------|
|128 WORD | SLOW WCS  |   FAST WCS    |
|HEADER   |           |               |
|---------|-----------|---------------|
```

```
|   |---------------------------------|
| 0| MICROCODE VERSION (8 BYTES ASCII) |
| 3|                                   |
|   |---------------------------------|
| 4| # OF WCS LOCATIONS (64 BIT WORDS) |
| 5|                                   |
|   |---------------------------------|
| 6| # OF LUT LOCATIONS (32 BIT WORDS) |
| 7|                                   |
|   |---------------------------------|
|10| SLOW WCS CHECKSUM                 |
|   |---------------------------------|
|11| FASTWCS CHECKSUM                  |
|   |---------------------------------|
```

Series 37, 37XP and 37 Micro WCS Table Format

Three entries (Entries 5, 6, and 7) used by Series 37, 37XP
and 37 Micro.

```
|---------|-----------|---------------|
|128 WORD | WCS       |   LUT         |
|HEADER   |           |               |
|---------|-----------|---------------|
```

```
|   |---------------------------------|
| 0| MICROCODE VERSION (8 BYTES ASCII) |
| 3|                                   |
|   |---------------------------------|
| 4| # OF WCS LOCATIONS (64 BIT WORDS) |
| 5|                                   |
|   |---------------------------------|
| 6| # OF LUT LOCATIONS (32 BIT WORDS) |
| 7|                                   |
|   |---------------------------------|
|10| WCS CHECKSUM                      |
|   |---------------------------------|
|11| LUT CHECKSUM                      |
|   |---------------------------------|
```

Store Tape Format

First Volume

```
|-------------------------------------|
|   | EOF                             |
|-------------------------------------|
|   | EOF                             |
|-------------------------------------|
| 0| "STORE/RESTORE LABEL-HP/3000."   |0  \
|15|                                  |13 |
|-------------------------------------|   |
|16| "VIIB"                          |14 |
|17|                                  |15 |
|-------------------------------------|   |
|20| PARTIAL FIRST FILE FLAG         |16 |
|-------------------------------------|
```

First Volume (Cont.)

```
|-------------------------------------|        |
|21| CHECKSUM                        |17     |
|-------------------------------------|        |
|22| DIRECTORY INDEX OF FIRST FILE   |18     |
|-------------------------------------|        | HEADER
|23|                                  |19     | 40 WORDS
|-------------------------------------|        |
|24|                                  |22     |
|-------------------------------------|        |
|27| VOLUME NUMBER                   |23     |
|-------------------------------------|        |
|30| DATE                            |24     | DATE:
|-------------------------------------|        |  0:7 last 2 digits
|31| TIME                            |25     |      of year
|32|                                  |26     |  7:9 Julian date
|-------------------------------------|        |
|33| TAPEBLOCKSIZE (#WORDS/BLOCK;DEF=4096) |27 | TIME:
|-------------------------------------|        |  25.(0:8) hours
|34|                                  |28     |     (8:8) minutes
| -|                                  | -     |  26.(0:8) seconds
|-------------------------------------|        |     (8:8) .1 secs.
|47|                                  |39     /
|-------------------------------------|
|   | EOF                             |
|-------------------------------------|
|   |          :                     |
|-------------------------------------|         \
| FILE NAME                          |\         |
|-------------------------------------| |TYP FILE | VOLUME
| GROUP NAME                         | | ENTRY   | DIRECTORY:
|-------------------------------------| |(12 WDS.)|  # ENTRIES
| ACCT. NAME                         |/          | DETERMINED
|-------------------------------------|           | BY TAPEBLOCK-
|   |          :                     |           / SIZE
|-------------------------------------|
|   | EOF                             |
|-------------------------------------|          \
| FILES  (SEPARATED BY "EOF'S")      |           | FILES
|-------------------------------------|          /
```

Subsequent Volumes

```
|------------------------------------------|
| 0| "STORE/RESTORE LABEL-HP/3000."      |0      \
|15|                                      |13      |
|------------------------------------------|        |
|16| "VIIB"                              |14      |
|17|                                      |15      |
|------------------------------------------|        |
|20| PARTIAL FIRST FILE FLAG             |16 FLAG=1:
|------------------------------------------|  1st FILE
|21| CHECKSUM                            |17 ON THIS
|------------------------------------------|  VOL IS A
|22| DIRECTORY INDEX OF FIRST FILE       |18 PARTIAL. | HEADER
|------------------------------------------|            | 40 WDS.
|23|                                      |19      |
|26|                                      |22      |
|------------------------------------------|        |
|27| VOLUME NUMBER                       |23      |
|------------------------------------------|        |
|30| DATE                                |24      |
|------------------------------------------|        |
|31| TIME                                |25      |
|32|                                      |26      |
|------------------------------------------|        |
|33| TAPEBLOCKSIZE                       |27      |
|------------------------------------------|        |
|34|                                      |28      |
|47|                                      |39      / NOTE: NO EOF.
|------------------------------------------|
|  -          .             -   |          \
|------------------------------------------|          |
| FILE NAME                    |\ TYPICAL  |
|------------------------------------------| FILE      | VOLUME
| GROUP NAME                   || ENTRY    | DIRECTORY
|------------------------------------------|/          |
| ACCT NAME                    |/          |
|------------------------------------------|          |
|  -          .             -   |          /
|------------------------------------------|
| EOF                          |
|------------------------------------------|
| <FILES>                      |          \ FILES
| (SEPARATED BY "EOF'S)        |          /
|------------------------------------------|
```

End of Volume

```
|------------------------------------------|
| <FILES>                      |          \
| (SEPARATED BY EOF'S)         |          | FILES
|------------------------------------------|          /
| EOF                          |
|------------------------------------------|
| 0| "STORE/RESTORE LABEL-HP/3000."      |0      \
|15|                                      |13      |
|------------------------------------------|        |
|16|                                      |14      |
|24|                                      |20      |
|------------------------------------------|        |
|25| FLAG: PRECEDING EOF MARKS FILE ENDED |21     | TRAILER
|------------------------------------------|        | 40 WDS.
|26| FLAG: PRECEDING EOF MARKS TAPESET ENDED |22   |
|------------------------------------------|        |
|27| VOLUME NO.                          |23      |
|------------------------------------------|        |
|30| DATE                                |24      |
|------------------------------------------|        |
|31| TIME                                |25      |
|32|                                      |26      |
|------------------------------------------|        |
|33|                                      |27      |
|47|                                      |39      /
|------------------------------------------|
| EOF                          |
|------------------------------------------|
| EOF                          |
|------------------------------------------|
| EOF                          |
|------------------------------------------|
```

Miscellaneous

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 tape marks. 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 tape mark 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.

Miscellaneous

HDR1: First header label. Required for each file and 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)
36/39	GENNUM	4	ALWAYS "0001"
40/42	VERSION	3	ALWAYS "00"
43/48	CREATION DATE	6	YEAR AND DAY WITHIN YEAR WHEN THE FILE WAS WRITTEN
49/53	EXPIRATION DATE	5	YEAR AND DAY WITHIN YEAR WHEN THE FILE MAY BE OVER-WRITTEN WITHOUT PERMISSION
54	ACCESSIBILITY	1	X230 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; it 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	BLKSIZE=RECSIZE?	1	YES="A", NO="B"
40/49	NOT USED	10	BLANKS
50/51	BUFFER OFF	2	ALWAYS "00"
52/80	NOT USED	29	BLANKS

IBM has a slightly different format which 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 then 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
|------------------------------------------------------------|
| 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           -
|                                                      |
|                                                      |
|------------------------------------------------------|
```

LCB Entry Format

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
|------------------------------------------------------|
| EXPIRATION DATE                                      |3
|------------------------------------------------------|
| FILE SEQUENCE NUMBER                                 |4
|------------------------------------------------------|
| CREATION DATE                                        |5
|------------------------------------------------------|
| BLOCK COUNT                                          |6
|                                                      |7
|------------------------------------------------------|
| REEL OF FILE                                         |10
|------------------------------------------------------|
| FILE NAME                                            |11
|                                                      |
-                                                      -
|                                                      |
|                               |----------------------|21
|                               |                      |22
| LOCKWORD                       |                     |23
|                                                      |24
|                                                      |25
|------------------------------------------------------|26
|                                                      |27
| VOLUME SET IDENTIFIER                                |
|                                                      |30
|------------------------------------------------------|31
|                                                      |32
| VOLUME IDENTIFIER                                    |
|                                                      |33
|------------------------------------------------------|
```

Type: 00 = No tape mounted
      01 = Unlabeled
      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.

VCB Entry Format

The VCB format is:

```
 0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| X | F | A |  POSITION  | U | SEQTYP| LBLTYP| L | M | R | B |0
|------------------------------------------------------|
| LDEV #                                               |1
|------------------------------------------------------|
| PIN                                                  |2
|------------------------------------------------------|
| FILE NUMBER (RFT INDEX)                              |3
|------------------------------------------------------|
| FILE SEQUENCE NUMBER                                 |4
|------------------------------------------------------|
| S | R |   | C |  DENSITY | U | REEL NUMBER           |5
|------------------------------------------------------|
| EXPIRATION DATE                                      |6
|------------------------------------------------------|
| REEL IN VOLUME SET                                   |7
|------------------------------------------------------|
| REEL OF FILE                                         |10
|------------------------------------------------------|
| FILE NAME                                            |11
|                                                      |
-                                                      -
|                                                      |
|                               |----------------------|21
|                               |                      |22
|                                                      |23
| LOCKWORD                                             |24
|                                                      |25
|------------------------------------------------------|26
|                                                      |27
| VOLUME SET IDENTIFIER                                |
|                                                      |30
|------------------------------------------------------|31
|                                                      |32
| VOLUME NAME                                          |
|                                                      |33
|------------------------------------------------------|
```

X: All Files Expired
F: Flush bit - operator did REPLY <pin>,0.
A: APPEND access
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: Reel switch 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.
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.
U: User logging warmstart recovery file access. (Set only during file open.)

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 unlabeled.

If the tape is unlabeled, AVREC reports to DEVREC that no further action is
required. If the tape is labeled, 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. Then 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 reel switching 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 Reel
switch, 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) = Z36

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'BKPT'EXT      |
                              |                     |
      FCB(18)                 |                     |
      |--|                    |---------------------|
      |  | --------------->   |-----                |
      |--|                    |---------------------|
                              |  |                  |
                              |  |                  |
                              |  -                  -
                              |  |                  |
                              |  |-\-\-\-\-\-\-\-\-\-\-\-
                              |  |                  |
                              |  |   BKPT'ENTRY'TAB |
                              |  |                  |
      SYS GLOBAL     ----->   |  |---------------------|
           14:15     |-----   |  |                  |
      |--------|-|-| |  |      |  |                  |
  Z25 |        |L|S| |  |      |  |                  |
      |--------|-|-| |  |      |                     |

      L = Table locked
      S = System break          -                  -
          points exist       |  |---------------------|
                             |---->                  |
                              |                     |
                              |---------------------|
                              |                     |
                              |                     |
                              |---------------------|
```
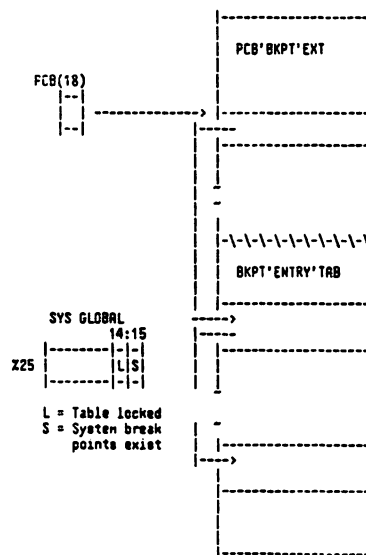
PCB Breakpoint Extension Table

```
|-----------------------|
|    # 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 minimum entry size is 7.

Active Entry

```
                        1 1 1 1 1 1
      0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
     |-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|
     |O|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                          |
     |-------------------------------|
   7 |  USERLABEL                \
     |---------------------------| |
     |                           | |
     -  CONDITION/COUNT          - |-> variable
     |                           | |
     |---------------------------| |
     |  COND DESCRIPTOR          | |
     |---------------------------| /
```

Active Entry (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
```

Active Entry (Cont.)

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 = ER
                     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.

```
                        1 1 1 1 1 1
          0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
          |-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|
      / 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    |
          |-------------------------------|
      /10| PTR TO MOST ACTIVE REQUEST    |HEAD
      |  |-------------------------------|
      |11| TRACE WORD                    |
 ENT2 |  |-------------------------------|
      |12|             0                 |
      |  |                               |DUMMY TIME
      \13|             0                 |----------
      |  |-|-------------|---------------|  |
      |14|A|  CODE       | INDEX OF NEXT |  |
      |  |-|-------------|---------------|  |
      |15| EG                            |  |
 ENT3 |  |-------------------------------|ASSIGNABLE
      |  | TIME TO SERVICE AFTER         |ENTRIES
      \  | REQUEST IN FRONT (UNIT= 100MS)|  |
          |-------------------------------|  |
      |  |                               |  |
      |  |                               |  |
          |-------------------------------|  |
```
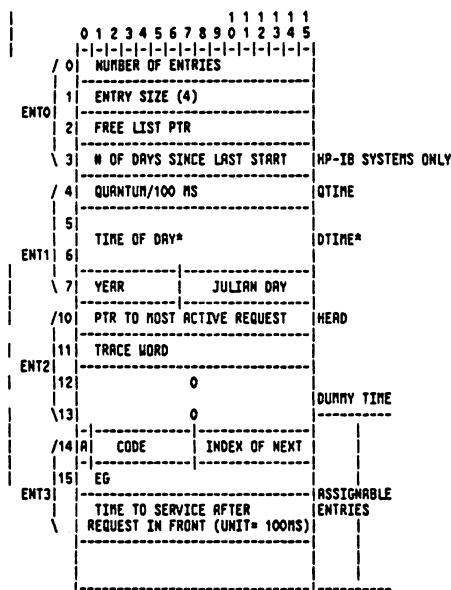
A: 0 if inactive request
   1 if active request

Timer Request List (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
X10	Port mask	Msg port timeout
X11	DITP	Block mode read timeout (30 secs)
X12	PCBB index to process	Watchdog timer for process
X13	Port DST	Port Procedure Timeout

The list of pending requests is kept ordered by time with later entries at the tail.

| X20-X37 | DITP | SIO device timeout: DIT8. (code_1 on expiration, cleared on Timereq. |
| X5/X6 | *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 it 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 journalize additions and modifications to MPE and subsystem files. The journal can reside on either disc or serial log files.

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 serial disc (or cartridge tape) log files the same as for tape files.

Design Structures

User Logging Table

```
|   ENTRY SIZE = X44 words
    DST X33
```

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

```
          #                         X
            |---------------------|
        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
            |---------------------|
   |   10|         .            |8
            |---------------------|
   |   45|         .            |37
            |---------------------|
```

```
    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
               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 the buffer file
               if serial log file (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|                       |0
            |-                   -|
          |   LOGGING           |
            |-                   -|
          |   IDENTIFIER        |
            |-                   -|
        4|                       |4
            |-                   -|
          |   BUFFER            |
            |-                   -|
          |   NAME             |
            |-                   -|
   |   10|                       |8
            |-                   -|
          |   FILE             |
            |-                   -|
          |   NAME             |
            |-                   -|
            |---------------------|
```

Typical Entry (Cont.)

```
   |   14|---------------------|12
            |-                   -|
          |   LOCK             |
            |-                   -|
          |   WORD             |
            |-                   -|
   |   20|---------------------|16
            |-                   -|
          |   GROUP            |
            |-                   -|
            |-                   -|
   |   24|---------------------|20
            |-                   -|
          |   ACCT             |
            |-                   -|
            |-                   -|
   |   30|  NUMBER OF USERS     |24
            |---------------------|
   |   31|  BUFFER DST NO       |25
            |---------------------|
   |   32|  LOG STATUS          |26
            |---------------------|
   |   33|  CURR AUTO | CURR TYPE|27
            |---------------------|
   |   34|  LOG DEV             |28
            |---------------------|
   |   35|  LOG PCB #           |29
            |---------------------|
   |   36|  SWITCH FLAG         |30
            |---------------------|
   |   37|  NEW AUTO | NEW TYPE  |31
            |---------------------|
   |   40|  ADDRESS OF          |32
            |-                   -|
          |   LOGGING BUFFER    |
            |---------------------|
   |   42|  SIZE OF             |34
            |-                   -|
          |   LOGGING BUFFER    |
            |---------------------|
   |   44|  FORWARD ENTRY PT    |36
            |---------------------|
   |   45|  BACKWARD ENTRY PT   |37
            |---------------------|
```

```
    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 flag 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 log-
ging 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                 -- |
|------------------------------------------------|---|---|
| ENTRY #2                                        |FPT|BPT|
|------------------------------------------------|---|---|
| ENTRY #3                                        |FPT|BPT|
|------------------------------------------------|---|---|
| ENTRY #4                                        |FPT|BPT|
|------------------------------------------------|---|---|
|                     .                                |
|                     .                                |
|                     .                                |
|------------------------------------------------|---|---|
| ENTRY #N                                        |FPT|BPT|
|------------------------------------------------|---|---|
|                                                      |
| BUFFER AREA                                          |
|                                                      |
| 4K WORDS                                             |
|                                                      |
|                                                      |
|------------------------------------------------------|
```

Communications Area

```
        |------------------------|
      0 |-                      -| 0
        |                        |
        |     LOGGING            |
        |-                      -|
        |     IDENTIFIER         |
        |-                      -|
        |                        |
        |------------------------|
      4 | SWITCH FLAG            | 4
        |------------------------|
      5 | CHANGE  | NEW TYPE     | 5
        |---------|--------------|
      6 | AUTO    | TYPE         | 6
        |---------|--------------|
      7 | BUFFER DST            | 7
        |------------------------|
     10 | LOG PIN               | 8
        |------------------------|
     11 | NUMBER OF USERS       | 9
        |------------------------|
     12 | MAX NUMBER OF USERS   |10
        |------------------------|
     13 | NEXT USER NUMBER      |11
        |------------------------|
     14 | SLEEP COUNT           |12
        |------------------------|
     15 | STATE                 |13
        |------------------------|
     16 | MSG                   |14
        |------------------------|
     17 | LOG MSG               |15
        |------------------------|
     20 | USER MSG              |16
        |------------------------|
     21 | LOG ERROR             |17
        |------------------------|
     22 | LOG DEVICE            |18
        |------------------------|
     23 | BUFFER SPACE          |19
        |------------------------|
     24 | USED SPACE IN BUFFER  |20
        |------------------------|
     25 | FILE SET NUMBER       |21
        |------------------------|
     26 | LOG                   |22
        |-                      -|
        | ADDRESS                |
        |------------------------|
     30 | INPUT                 |24
        |-                      -|
        | RECORD                 |
        |------------------------|
```

Communications Area (Cont.)

```
     |------------------|              |------------------|
  32 | FILE         |26            56 | OLD'NUM'EXTENT |46
     |-             -|                 |------------------|
     | SIZE         |            57                    |47
  34 | FILE         |28            60 | IN USE HEAD PTR|48
     |-             -|                 |------------------|
     | SPACE        |            61 | FREED HEAD PTR |49
     |------------------|              |------------------|
  36 | TOTAL        |30            62 | FIRST FILE     |50
     |-             -|                 |-               -|
     | RECORDS      |                 | CREATION TIME  |
  40 | MAX          |32            64 | FIRST CREATION DATE|52
     |-             -|                 |------------------|
     | SIZE         |            65 | F'TYPE         |53
  42 | LAST EXTENT  |34            66 | P'TYPE         |54
     |------------------|              |------------------|
  43 | EXTENT       |35            67 | C'TYPE         |55
     |------------------|              |------------------|
  44 |              |36            70 | N'TYPE         |56
     |-             -|                 |------------------|
     | RESOURCE1    -|
     |-             -|
     |              |
     |------------------|
  50 |              |40
     |-             -|
     | RESOURCE2    |
     |-             -|
     |              |
     |------------------|
  54 | OLD LIMIT    |44
     |-             -|
     |              |
     |------------------|
```

Communications Area (Cont.)

```
|   |---------------------|   |     |---------------------|     |
| 69| FIRST              |57 | 201 |                     |129  |
|   |- FILE              - | |     |-                   - |     |
|   |                     | |     |                     |     |
|   |-                   - | | 206 | REC'S IN           |134  |
|   |                     | |     | PREVIOUS FILE       |     |
|   |-                   - | | 210 | F.S. ERROR         |136  |
|   |                     | |     |---------------------|     |
|   |                     | | 211 | U.L. ERROR         |137  |
|   |                     | |     |---------------------|     |
|   |                     | | 212 | HEAD PIN           |138  |
| 87| PREVIOUS           |75 | 213 |                     |139  |
|   |- FILE              - | | 214 | RESOURCE3          |140  |
|   |                     | |     |                     |     |
|   |-                   - | |     |-                   - |     |
|   |                     | |     |                     |     |
|   |                     | |     |-                   - |     |
|135| CURRENT            |93 |     |---------------------|     |
|   |- FILE              - |
|   |                     |
|   |-                   - |
|   |                     |
|157| NEXT FILE          |111|
|   |                     |
|   |-                   - |
|   |                     |
|   |---------------------|
```

LOGID	=	BLOGBUFF(0)
SWITCH'	=	LOGBUFF(4)
RCHANGE	=	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	=	LOGBUFF(36)
RESOURCE2	=	LOGBUFF(40)
UHEAD	=	LOGBUFF(48)
FHEAD	=	LOGBUFF(49)
FIRST'C'TIME	=	DLOGBUFF(50)
FIRST'C'DATE	=	LOGBUFF(52)
F'TYPE	=	LOGBUFF(53)
P'TYPE	=	LOGBUFF(54)
C'TYPE	=	LOGBUFF(55)
N'TYPE	=	LOGBUFF(56)
FIRST'FILE	=	BLOGBUFF(57)
PREVIOUS'FILE	=	BLOGBUFF(75)
CURRENT'FILE	=	BLOGBUFF(93)
NEXT'FILE	=	BLOGBUFF(111)
RECSIN'PREV	=	DLOGBUFF(67)

FSERR'CODE	=	LOGBUFF(136)
ULERR'CODE	=	LOGBUFF(137)
HEAD'CHANGE'PIN	=	LOGBUFF(138)
RESOURCE3	=	LOGBUFF(140)
NOT'SAFE'TO'STOP	=	LOGBUFF(144)

LOGID   - The name of the logging process.

SWITCH' - True if log file switch is in progress.

CHANGE - True if log file name, such that changelog is allowed
          (i.e., first file in the set name filename 001).

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.

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
          and is 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 - True if error condition during changelog.

LOGDEV - The logical device number of the current extent
          of the disc file 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.

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. (Current 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).
          (Current file.)

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). (Total records written to all log files in the set.)

MAXFSPACE - The total file size, in records, minus 1. (Need that last record to post close information.) (Current file.)

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 buffer area and buffer information in the communications area). 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

RESOURCE2 - Use for locking file information and messages in the communications area.

OLD'LIMIT - The number of records in the last disc log file.

OLD'NUM"EXTENT-The number of extents in the last disc log file.

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)

FIRST'C'TIME-First file creation time.

FIRST'C'DATE-Firest file creation date.

F'TYPE - First log file type.

P'TYPE - Previous log file type.

C'TYPE - Current log file type.

N'TYPE - Next log file type.

FIRST'FILE - First log file in the log sequence.

PREVIOUS'FILE-Previous log file in the log sequence.

CURRENT'FILE -Current log file in the log sequence.

NEXT'FILE - Next log file in the log sequence.

RECS'IN'PREV- Total number of records in all of the previous file in the log file set.

FSERR'CODE - File system error encountered upon changelog.

ULERR'CODE - User logging error encountered upon changelog.

HEAD'CHANGE'PIN-PCB index of process waiting for :CHANGELOG command to flush. Note only one process waiting at a time.

RESOURCE3 - Use for locking user entry area and pointer information about the user entries in the communications area.

NOT'SAFE'TO'STOP-If it is set, then do not process the Stoplog until cxchangelog resets the bit.

Typical Logbuff Entry

```
          0|--------------------|0
           |-                  -|
           |      USER           |
           |-                  -|
           |      NAME           |
           |-                  -|
          4|--------------------|4
           |-                  -|
           |     GROUP           |
           |-                  -|
           |     NAME            |
           |-                  -|
    |    10|--------------------|8
           |-                  -|
           |- ACCOUNT          -|
           |-                  -|
           |- NAME             -|
           |--------------------|
    |    14| USER PCB #         |12
           |--------------------|
    |    15| OPENLOG COUNT      |13
           |--------------------|
    |    16| WAIT STATE         |14
           |--------------------|
    |    17| ERROR CODE         |15
           |--------------------|
    |    20| LOG NUMBER         |16
           |--------------------|
    |    21| SUBSYSTEM CODE     |17
           |--------------------|
    |    22|    TOTAL          -|18
           |-                  -|
           |   RECORDS           |
           |--------------------|
           |-                  -|
           |-                  -|
           |--------------------|
    |    27| FRWD ENTRY PTR     |23
           |--------------------|
    |    30| BKWRD ENTRY PTR    |24
           |--------------------|
```

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
USTATE	=	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).

USTATE - 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 = OK.

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 = Z41 words
DST Z41

Table containing an entry for each potential logging process. Entries are added via :GETLOG and released via :RELLOG.

Entry #0

```
          #                    Z
          |--------------------|
        0 |                    |0
          |--------------------|
        1 |MAX NUMBER OF ENTRIES|1
          |--------------------|
        2 |                    |2
          |--------------------|
        3 |                    |3
          |--------------------|
        4 |     ENTRY SIZE     |4
          |         .          |
          |         :          |
       40 |         :          |32
          |--------------------|
```

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        -|
            |-                  -|
         10 |--------------------|8
            |-                  -|
            |   FILE             |
            |-                  -|
            |   NAME             |
         14 |--------------------|12
            |-                  -|
            |   FILE             |
            |-                  -|
            |   LOCKWORD        -|
         20 |--------------------|16
            |-                  -|
            |   FILE             |
            |-                  -|
            |   GROUP            |
         24 |--------------------|20
            |-                  -|
            |   FILE             |
            |-                  -|
            |   ACCOUNT          |
            |-                  -|
            |--------------------|
```

Typical Entry (Cont.)

```
         30 |--------------------|24
            |-                  -|
            |   USER'S           |
            |-                  -|
            |   NAME             |
         34 |--------------------|28
            |-                  -|
            |   USER'S           |
            |-                  -|
            |   ACCOUNT          |
         40 |   LOG TYPE         |32
            |--------------------|
```

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 password 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 lockword 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.
```
          (0:1) = Auto changelog allowed
          (1:1) = No Auto on :GETLOG, :ALTLOG
          (2:7) = Previous type
          (9:7) = New type
                0 = Disc log file
                1 = Tape log file
                2 = Serial disc log file
                3 = Cartridge tape log 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
|----|----|----|----|----|----|-----|-------|----|----|
|RECN|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
|----|----|----|----|----|----|----|---------------|
|RECN|CKSUM|CODE|TIME|DATE|LOG#|LEN|   USER AREA    |
|----|----|----|----|----|----|----|---------------|
```

LOG RECORD AT CLOSELOG

```
0  2   3   4   6   7   11  12       24  25   127
|----|----|----|----|----|----|-----|-------|----|----|
|RECN|CKSUM|CODE|TIME|DATE|LOGID|LOG#| CREATOR|PCB |    |
|----|----|----|----|----|----|-----|-------|----|----|
```

CRASH MARKER

```
0  2   3   4   6   7                          127
|----|----|----|----|----|---------------------|
|    |    |    |    |    |                     |
|----|----|----|----|----|---------------------|
```

```
| RECN|CKSUM|CODE |TIME|DATE|                                  |
|-----|-----|-----|----|----|----------------------------------|
```

HEADER RECORD (START/RESTART)

```
0   2     3     4    6    7    11                          127
|----|-----|-----|----|----|-----|---------------------------|
|    |     |     |    |    |     |                           |
|RECN|CKSUM|CODE |TIME|DATE|LOGID|                           |
|----|-----|-----|----|----|-----|---------------------------|
```

TRAILER RECORD (STOP)

```
0   2     3     4    6    7    11                          127
|----|-----|-----|----|----|-----|---------------------------|
|    |     |     |    |    |     |                           |
|RECN|CKSUM|CODE |TIME|DATE|LOGID|                           |
|----|-----|-----|----|----|-----|---------------------------|
```

NULL RECORD

```
0   2     3     4    6    7                               127
|----|-----|-----|----|----|-------------------------------|
|    |     |     |    |    |                               |
|RECN|CKSUM|CODE |TIME|DATE|                               |
|----|-----|-----|----|----|-------------------------------|
```

BEGIN TRANSACTION MARKER

```
0   2     3     4    6    7   8   9                       127
|----|-----|-----|----|----|---|---|-----------------------|
|    |     |     |    |    |   |   |                       |
|RECN|CKSUM|CODE |TIME|DATE|LOGN|LEN|    USER AREA          |
|----|-----|-----|----|----|---|---|-----------------------|
```

END TRANSACTION MARKER

```
0   2     3     4    6    7   8   9                       127
|----|-----|-----|----|----|---|---|-----------------------|
|    |     |     |    |    |   |   |                       |
|RECN|CKSUM|CODE |TIME|DATE|LOGN|LEN|     USER AREA         |
|----|-----|-----|----|----|---|---|-----------------------|
```

CHANGELOG RECORD

```
0   2     3     4    6    7   8
|----|-----|-----|----|----|---|-----------------------------|
```

```
|    |     |     |    |    |     |                             (
|RECN |CKSUM|CODE|TIME|DATE|LOGID|                            |
|-----|-----|----|----|----|-----|---------------------------|
```

CHANGELOG RECORD (Cont.)

```
11   12   14    15    33   34    52   53    72   127
|----|----|-----|-----|----|-----|----|-----|-----|---|
|    |    |     |     |    |     |    |     |     |   |
/ SEQ|C-TIME|C-DATE|F-FILE|F-TYPE|P-FILE|P-TYPE|C-FILE|C-TYPE|
|----|----|-----|-----|----|-----|----|-----|-----|---|
```

Note: If CODE = 12, P-File = Previous file in set.
       If CODE = 13, P-FILE = Next file in set.

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
    12   Changelog record in new file
    13   Changelog record in old file
  SPACE  NULL record
```

DATA FIELDS OF LOG RECORDS

RECN	=	DOUBLE INTEGER
CKSUM	=	INTEGER
CODE	=	INTEGER
TIME	=	DOUBLE (from intrinsic CLOCK)
DATE	=	INTEGER (from intrinsic CALENDAR)
LOGID	=	ASCII
LOGN	=	INTEGER
LEN	=	INTEGER
USERAREA	=	ASCII
CREATOR	=	ASCII
PCB	=	INTEGER
C-DATE	=	INTEGER
C-TIME	=	DOUBLE
F-FILE-NAME	=	ASCII
P-FILE-NAME	=	ASCII
C-FILE-NAME	=	ASCII
F-TYPE	=	INTEGER
P-TYPE	=	INTEGER
C-TYPE	=	INTEGER
SEQ	=	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 = X73 (59)

```
--|---|------------------------------------|
  | 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| CLASS 15 STATISTICS XDS BANK       |MEASPROC-
  |   |                                    |XDSBANK
  |---|------------------------------------|
  |22| CLASS 15 STATISTICS XDS BASE       |MEASPROC-
  |   |                                    |XDSBASE
  |---|------------------------------------|
  |23| CLASS 15 STATISTICS XDS NUMBER     |MEASPROC-
  |   |                                    |XDSNUM
  |---|------------------------------------|
  |24| CLASS 14 STATISTICS XDS BANK       |
  |---|------------------------------------|
  |25| CLASS 14 STATISTICS XDS BASE       |
  |---|------------------------------------|
```

MEASINFOTAB (Cont.)

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 12 STATISTICS 2ND XDS BANK
41	CLASS 12 STATISTICS 2ND XDS BASE
42	CLASS 12 STATISTICS 2ND XDS NUM.
43	CLASS 15 STATISTICS 2ND XDS BANK
44	CLASS 15 STATISTICS 2ND XDS BASE
45	CLASS 15 STATISTICS 2ND XDS NUM.

** As of Release 23, all pin #s > 629 for classes 12 and 15 will
appear in the 2ND set of extra data segments.

MEASINFOTAB (Cont.)

reserved for measurement interface		
50	CLASS 0 ENABLED COUNT	CLASS 1 ENABLED 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 shared clock interface user 62		
63		
64		
65		
66		
67		

MEASINFOTAB (Cont.)

	70	M FLAG A	
shared	71	XDS1	
clock	72	XDS2	
interface	73	DCOUNT	
cells	74	DLIMIT	
	75	TCOUNT	
	76	TLIMIT	
	77	DLABEL	
	100	MONITOR BUFFER INDEX	SMONIDX
	101	MEAS BUFFER	MEASBUF0
	102	MEAS BUFFER INDEX	MEASIDX
reserved for event logging	103	MEAS ENABLED FLAGS	MEASMSK0
	104	MEAS ENABLED FLAGS	MEASMSK1
	105	MEAS BUFFER BANK	MEASBUFBANK
	106		
	.		
	.		
	116		
	117		

M: Interrupt has missed due to last interrupt handling.

A: Current interrupt handling active.

Security DST Layout

System Global Security DST

DST # in SYSGLOB extension X30.

X	Table 1	
0	LENGTH OF FIRST TABLE	---\
1	USE COUNT	
2	CURRENT PASSWORD ENCRYPTION (ON/OFF)	
3	MINIMUM PASSWORD LENGTH (# OF CHARS)	
4	MAX # OF INVALID LOGON ATTEMPTS PER DEV.	
5	PASSWORD PROMPT REQUIRED OPTION (ON/OFF)	
6	UDC FAILURE TERMINATION OPTION (ON/OFF)	
7	GENERIC LOGON INTERFACE ERROR MSG (ON/OFF)	
10	FOPEN FAILURE LOGGING ONLY OPTION	
11	IDLE SESSION TIME-OUT IN SECONDS	
12	SECURITY DOWN TIME-OUT IN SECONDS	
13	programmatic access warning flag	
14	Password expiration interval in days	
15	Next global password expiration date	
16	W|C| | | | | | Number of days to warn	
17	Embedded password disallowed for jobs	
20	Cross streaming disallowed for jobs	
21	Stream privilege for bypass password	
22	Assurance of logging	
23	File maximum protection on creation	

W = Warning flag set in user entry
C = SECURITYPROC to clear the warn flag

System Global Security DST (Cont.)

```
                    Table 2
   |---------------------------------------------|
  0| LENGTH OF SECOND TABLE - LOGON ATTEMPTS      |<---/
   |---------------------------------------------|
  1| LOGON ATTEMPTS COUNT - 1 WORD PER DEVICE     |----\
   |      INDEXED BY DEVICE NUMBER                |
  .|                                             |
  .|                                             |
   |---------------------------------------------|


                    Table 3
   |---------------------------------------------|
  0| COMMAND TABLE LENGTH (n)                     |<---/
   |---------------------------------------------|
  1| COMMAND # 1 INFO                             |----\
   |---------------------------------------------|
  2| COMMAND # 2 INFO                             |
   |---------------------------------------------|
   |                    .                        |
   |                    .                        |
   |---------------------------------------------|
  n| COMMAND # n INFO                             |
   |---------------------------------------------|


                    Table 4
   |---------------------------------------------|
  0| LENGTH OF TABLE 4 - DEVICE PASSWORDS         |<---/
   |---------------------------------------------|
  1| LOGICAL DEVICE NUMBER                        |----\
   |---------------------------------------------|
  2| PASSWORD FOR LOGON PASS. PROMPT (8 CHARS)    |
  3|                                             |
  4|                                             |
  5|                                             |
   |---------------------------------------------|
   |                    .                        |
   |                    .                        |
   |---------------------------------------------|
   | LOGICAL DEVICE NUMBER                        |
   |---------------------------------------------|
   | PASSWORD FOR LOGON   (8 characters)          |
   |---------------------------------------------|
   | 0 (END OF SECURITY DST)                      |<---/
   |---------------------------------------------|
```

Command Info Entry

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

P = Programmatically execution disabled. (1=disabled, 0=enabled)
E = Execution of this command disabled.   (1=disabled, 0=enabled)
L = Logging enabled for this command.   (1=enabled, 0=disabled)

```
EQUATE
  SEC'ENCRYPTION'WORD        = 2,
  SEC'PASS'LEN'WORD          = 3,
  SEC'NUM'ATTEMPTS'WORD      = 4,
  SEC'REQUIRED'PROMPT'WORD   = 5,
  SEC'UDC'TERMINATION'WORD   = 6,
  SEC'GENERIC'MSG'WORD       = 7,
  SEC'FOPEN'LOGGING'WORD     = 8,
  SEC'SESSION'TIMEOUT'WORD   = 9,
  SEC'DOWN'TIMEOUT'WORD      =10,
  SEC'PROG'WARN'WORD         =11,
  SEC'PW'AGING'WORD          =12,
  SEC'EXP'DATE'WORD          =13,
  SEC'WARN'DAY'WORD          =14,
  SEC'JOB'EMBED'PW'WORD      =15,
  SEC'CROSS'STREAM'WORD      =16,
  SEC'STREAM'PRIV'WORD       =17,
  SEC'INSURE'LOG'WORD        =18,
  SEC'MAX'PROTECT'WORD       =19;

EQUATE
  GLOBAL'OPTION'TAB          = 1,
  ATTEMPT'COUNT'TAB          = 2,
  COMMAND'INFO'TAB           = 3,
  DEVICE'PASSWORD'TAB        = 4;

EQUATE
  TAB1'LEN                   = 25,
  TAB2'LEN                   = 1025,
  TAB3'LEN                   =400;
<<TAB4'LEN is variable>>
```

Security Table - DACD (Device Access Control Definition)

```
                   0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
                  |-|-|-|-|-|-|-|-|-|-|--|--|--|--|--|--|
 |FLABEL-------    FILE LABEL (if DACDDST.PUB.SYS)
 |              / |-------------------------------------|
 |             /  |       TABLE SIZE(words)             |
 |             \  |-------------------------------------|
 |HEADER        \ |       DST NUMBER                    |
 |             / \|-------------------------------------|
 |            /   |       NUMBER OF ENTRIES             |
 |            \   |-------------------------------------|
 |             \  |       UNUSED                        |
 |            /   |-------------------------------------|
 |           /    | VTAB INDEX  |      HODA             |
 |           \    |-------------------------------------|
 |            \   |       LODA                          |
 |LDEV 1       \  |-------------------------------------|
 |            /   | PSEUDO EXTENT SIZE(sectors)         |
 |           /    |-------------------------------------|
 |           \    |       UNUSED                        |
 |            \   |-------------------------------------|
 |           /    | VTAB INDEX  |      HODA             |
 |          /     |-------------------------------------|
 |          \     |       LODA                          |
 |LDEV 2     \    |-------------------------------------|
 |          /     | PSEUDO EXTENT SIZE(sectors)         |
 |         /      |-------------------------------------|
 |         \      |       UNUSED                        |
 |          \     |-------------------------------------|
 |         /      | VTAB INDEX  |      HODA             |
 |        /       |-------------------------------------|
 |        \       |       LODA                          |
 |LDEV NNN \      |-------------------------------------|
 |        /       | PSEUDO EXTENT SIZE(sectors)         |
 |       /        |-------------------------------------|
 |       \        |       UNUSED                        |
 |        \       |-------------------------------------|
 |       /        | VTAB INDEX  |      HODA             |
 |      /         |-------------------------------------|
 |      \         |       LODA                          |
 |LDEV 999 \      |-------------------------------------|
 |       /        | PSEUDO EXTENT SIZE(sectors)         |
 |      /         |-------------------------------------|
 |      \         |       UNUSED                        |
 |       \        |-------------------------------------|
```

Device ACD's are pointed to by the Device ACD table. The
Device ACD table is stored in DACDDST.PUB.SYS. When the
system is brought up, this file will be copied to a data
segment for faster access. The DACD is indexed by ldev number.

JSECDST Table (DST X75)

```
     DST = 61 = X75
     SIR = 15 = X17 (JMAT SIR IS USED)

         0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
        |-|-|-|-|-|-|-|-|-|-|--|--|--|--|--|--|
    X0  | Maximum size       |   current size   |   (entry 0)
        |-------------------------------------------|
     1  | entry size (X46 words)                    |
        |-------------------------------------------|
     2  | offset to first entry (X46)               |
        |-------------------------------------------|
     .  |                                           |
     .  | Reserved                                  |
        |                                           |
    45  |                                           |
        |-------------------------------------------|
    X46 | TY |     JOB/SESSION NUMBER               |   X0 (entry 1)
        |-------------------------------------------|
     1  | 2nd half of J/S number unused in MPE V/E  |
        |-------------------------------------------|
     2  | TY |     Initiator Job/Session number     |
        |-------------------------------------------|
     3  | 2nd half of J/S number unused in MPE V/E  |
        |-------------------------------------------|
     4  | Initiator job/session name (4 words)      |
        |                                           |
        |-------------------------------------------|
    10  | Initiator user name (4 words)             |
        |                                           |
        |-------------------------------------------|
    14  | Initiator account name (4 words)          |
        |                                           |
        |-------------------------------------------|
    20  | Initiator's logon ldev number             |
        |-------------------------------------------|
    21  | Initiating date                           |
        |-------------------------------------------|
    22  | Initiating time (2 words in CLOCK format) |
        |-------------------------------------------|
     .  |                                           |
   113  |                                           |   45
        |-------------------------------------------|
```

Information on who, when and where a job is streamed will be displayed
in the job's $STDLIST. This info will be put in the Job Security

Master Table at job creation time.

The JSEC DST is similar to the JMAT, there is one per system. JSEC will have the same entry size and total size as the JMAT. The JMAT index will be used to allocate, access and deallocate JSEC entries. INITIAL will create or recover the JSEC the same way it does the JMAT.

The JSEC is preserved on disc in the file JSECDST.PUB.SYS. INITIAL will create DST X75 using the contents of JSECDST.PUB.SYS.

Access Control Definition

An Access Control Definition (ACD) has the following format:

```
||------------------------------|
|| Number of ACD entries        |------------------|
||------------------------------|                  |
|| ACD checksum                 |                  |ACD HEADER
||------------------------------|                  |
|| ACD release number           |-------|          |
||------------------------------|       |          |
|| ACD update date              |ACD    |          |
||------------------------------|VERSION |
|| ACD update time (2 words)    |       |
||------------------------------|-------|
|| Account name (4 words)       |-------|
||------------------------------|       |
|| User name    (4 words)       |ACD ENTRY 1
||------------------------------|       |
|| Security modes               |       |
||------------------------------|       |
|| Account name (4 words)       |       |
||------------------------------|       |
|| User name    (4 words)       |ACD ENTRY 2
||------------------------------|       |
|| security modes               |       |
||------------------------------|
|.                              .
|.                              .
|.                              .
||------------------------------|-------|
|| Account name (4 words)       |       |
||------------------------------|       |
|| User Name    (4 words)       |ACD ENTRY 20
||------------------------------|       |
|| Security modes               |       |
||------------------------------|-------|
```

An ACD consists of the ACD HEADER followed by a maximum of 20 entries.

The ACD HEADER has 3 components: NUMBER OF ACD ENTRIES, ACD CHECKSUM and ACD VERSION. The ACD VERSION includes the ACD RELEASE NUMBER, the ACD UPDATE DATE and the ACD UPDATE TIME.

NUMBER OF ACD ENTRIES: Number of entries currently in this ACD.

ACD CHECKSUM: A number representing the EXCLUSIVE OR of all the words that comprise the entries in the ACD.

ACD RELEASE NUMBER: A number representing the current ACD software used to create the ACD. (MPE V/E VD4 = 5)

ACD UPDATE DATE: Date when the ACD was last modified(CALENDAR format).

ACD UPDATE TIME: Time when the ACD was last modified(CLOCK format).

There can be a maximum of 20 entries. Each entry consists of an ACCOUNT NAME, USER NAME, and the SECURITY MODES granted to the specified user. Wild cards can be used instead of ACCOUNT NAME and USER NAME. The only valid wild card user specifications are:

    @.ACCOUNT
    @.@

"@." is represented internallt with the character "`".

Entries are sorted as per the following example:

    SAM.ACCTING
    TOM.ACCTING
    @.ACCTING
    ROSE.FINANCE
    @.FINANCE
    @.@

Each entry consists of:

ACCOUNT NAME: The account name specified by the creator. Upshifted, with trailing blanks added.
USER NAME: THE USER name specified by the creator. Upshifted, with trailing blanks added.
SECURITY MODES: This is the word indicating the access/permission the user is granted. It is 2 bytes (1 word).

```
| 0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 |
|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
|R | W| X| A| L| UNUSED | Z|   UNUSED      | N|
|------------------------------------------------|
```

Note: Z = Permission to read ACD, N = No access

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, such as the file label, file control block, and the access control block.

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 (X77)     |
|----------------------------|
| 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 message 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| EU    | 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.

EU (2:1) - Set for the last record written before the file label EOF
           was updated.

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.

```
|    0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
|   |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
-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|                                                 |-1
  |--|---------------------------------------------|
 0|  | SIZE OF THE ACB INCLUDING BUFFERS (WORDS)   |0
  |--|---------------------------------------------|
 1| FILE NUMBER                                     |1  *
  |------------------------------------------------|
 2| FILE NAME                                       |2  *
  |                                                 |   *
  |------------------------------------------------|
 6| FOPTIONS                                        |6  *
  |------------------------------------------------|
 7| AOPTIONS                                        |7  *
  |------------------------------------------------|
10| RECORD SIZE (BYTES)                             |8  *
  |------------------------------------------------|
11| BLOCK SIZE (WORDS)                              |9  *
  |------------------------------------------------|
12|                                                 |10
  |------------------------------------------------|
13| CARRIAGE CONTROL CODE (WRITERS)                 |11 *
  |------------------------------------------------|
14|                                                 |12 *
  |------------------------------------------------|
15|                                                 |13
  |------------------------------------------------|
```

Message Access Control Block (Cont.)

```
|    0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
|   |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
16| ERROR CODE                                      |14 *
  |------------------------------------------------|
17| TRANSMISSION LOG (UNITS SAME AS LAST RD/WRITE   |15 *
  |------------------------------------------------|
20| TOTAL NUMBER OF UNREAD RECORDS (INCLUDES        |16
  |------------------                               |
21| OPENS AND CLOSES)                               |17
  |------------------------------------------------|
22| BLOCK NUMBER OF THE FILE'S TAIL (RELATIVE TO    |18
  |------------------                               |
23| THE START OF FILE BLOCK)                        |19
  |------------------------------------------------|
24| LOGICAL RECORD TRANSFER COUNT                   |20
  |------------------                               |
25|                                                 |21
  |------------------------------------------------|
26| PHYSICAL BLOCK TRANSFER COUNT                   |22
  |------------------                               |
27|                                                 |23
  |------------------------------------------------|
30| DST REL ADDR OF READ HEADER                     |24
  |------------------------------------------------|
31| DST REL ADDR OF WRITE HEADER                    |25
  |------------------------------------------------|
32| FCB DST                                         |26
  |------------------------------------------------|
33| FCB VECTOR TABLE OFFSET                         |27
  |------------------------------------------------|
34| SHARE COUNT (NUMBER OF LACB'S)                  |28
  |------------------------------------------------|
35| ACCESS CLASS, STATUS, ETC.                      |29
  |------------------------------------------------|
36| LOGICAL DEVICE NUMBER                           |30
  |------------------|--------------|--------------|
37|                  | WRT BUF INX  | # BUF - 1    |31
  |------------------|--------------|--------------|
40| DST RELATIVE ADDRESS OF NEXT READ RECORD        |32
  |------------------------------------------------|
41| SIZE OF THE BUFFER (WORDS)                      |33
  |------------------------------------------------|
42| SPARE                                           |34
  |------------------------------------------------|
43| FMAVT INDEX                                     |35
  |------------------------------------------------|
44| NUMBER OF READ LACB'S                           |36
  |------------------------------------------------|
```

Message Access Control Block (Cont.)

	0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15	
45	TYPE AND DISPOSITION	37
46	ACCESS MASK / RECORDS PER BLOCK	38
47	MISC. MSG FILE FLAGS	39
50	0 / # RD BUF / # WT BUF / ER|QW|M|C|D|S|F	40
51	NUMBER OF FREE WORD IN THE CURRENT FREE RECORD	41
52	NUMBER OF FREE RECORDS	42
53		43
54	NUMBER OF NONDATA RECORDS IN THE FILE	44
55		45
56	DST RELATIVE ADDRESS OF THE NEXT WRITE RECORD	46
57	#OPEN RECORDS / # READ REQUESTS	47
60	LAST READ ERROR / LAST WRITE ERROR	48
61	HEAD RECORD'S TYPE	49
62	HEAD RECORD'S WRITER ID	50
63	HEAD RECORD'S FLAGS	51
64	DST REL ADDRESS OF THE PACB	52
65	DST REL ADDRESS OF THE LACB	53
66	DST RELATIVE ADDRESS OF THE STACK ACB	54
67	STACK DST RELATIVE ADDRESS OF DB	55
70	TARGET AREA'S DST NUMBER	56
71	RESERVED FOR CALLING PARAMETERS	57
72		58
73		59

Message Access Control Block (Cont.)

	0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15	
74	RESERVED FOR THE STACK MARKER FROM FILE SYSTEM	60
75	INTRINSICS	61
100	USER'S SOFT INTERRUPT PLABEL	64 *
101	# OF SECONDS TO WAIT ON BOUNDARY CONDITION	65 *
102	O|EX|ND|VR|BT|IM|C|W| CARRIAGE CONTROL	66 *
103	REPLY PORT (BASIC IPC PORT)	67 *
104	WRITER ID	68 *
105	NOWAIT WRITER RECORD BUFFER ADDRESS	69 *
106	NOWAIT WRITER RECORD BUFFER DST	70 *
107	NOWAIT WRITER BUFFER ENTRY NUMBER	71 *
110	NO WAIT I/O RESULTANT ERROR CODE	72 *
111	NO WAIT I/O RESULTANT TRANSMISSION LOG	73
112	NO WAIT I/O FREAD TARGET DST	74 *
113	NO WAIT I/O FREAD TARGET ADDRESS 113*	75 *
114	WRITE WAIT QUEUE (BASIC IPC PORT)	76
115	READ WAIT QUEUE (BASIC IPC PORT)	77
116	RECORD SIZE & OVERHEAD	78
117	HEAD RECORD'S LENGTH IN BYTES	79
120	WRITER ID	80
121	LOCAL FLAGS	81
122	TARGET DST NUMBER	82
123	DST RELATIVE ADDRESS OF TARGET AREA	83
124	LENGTH OF TARGET AREA	84

Message Access Control Block (Cont.)

	0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15	
125	WRITER'S REPLY PORT. 0=USING ACB COMPLTN AREA	85
126	WAITING PROCESS'S PIN	86
127	WAITING PROCESS'S FILE NUMBER	87
130	WRITER'S SOFT INTERRUPT PLABEL	88
131	RESULTANT ERROR CODE	89
132	RESULTANT TRANSMISSION LOG	90
133	RESULTANT WRITE ID	91
134	DST REL ADDRESS OF FIRST BUFFER	92
135	DST REL ADDRESS OF BUFFER TWO	93

.
.
.

* Value is private to a particular accessor.

Message Access Control Block (Cont.)

Word	Field	Description
%50		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	0 - the start of file is block zero
	(15:1) F	0 - the ACB buffers have not been filled
%102		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 first record.
	(4:1) BT	0 - transmission log should be expressed in words. 1 - " " " " " " bytes.
	(5:1) IM	1- only user mode procedures can be soft interrupted.
	(6:1) C	- no wait completion message is in LACB area.
	(7:1) W	1 - wait disabled and just opened.
	(8:8) CAR CTL	- carriage control character to be used for the writer's record (a value of one indicates no carriage control character).

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) # rec,	(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) Wfree,	(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 thirty-five hundred open ports and outstanding messages, therefore, ports and message blocks are not 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

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 (i.e., 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 #| -------
 + X100               |----------|       |
                                         |
                                         |
                      |----------------| |
                      |                |<-----|
PORT DATA SEGMENT     | GLOBAL AREA    |
 |                    |                |
                      |----------------|
                      |                |
                      | REMAINDER IS   |
                      | COMPOSED OF    |
                      | "BLOCK SIZE"   |
                      | CHUNKS.        |
                      |----------------|
```
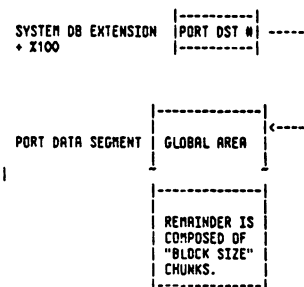
The chunks are a combination of free entries, ports, message queue entries, and timer list entries.

Port With Two Outstanding Messages

```
|------------|        |------------|        |------------|
|            | |----->|            | |----->|            |
|   PORT     | |      |   MQE 1    | |      |   MQE 2    |
|            | |      |            | |      |            |
|------------|        |------------|        |------------|
```
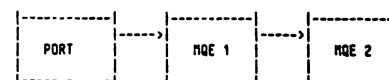
Port Number

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

```
    |-----------------------------------------------|
100| PORT DATA SEGMENT NUMBER                       |64
    |-----------------------------------------------|
101| 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| PIN OR PORT OWNER                  |3
  |--|-------|-------------------------------------|
 4| SOFT INTERRUPT FILE NUMBER                    |4
  |-----------------------------------------------|
 5| NUMBER OF MQES IN THE PORT'S QUEUE            |5
  |-----------------------------------------------|
 6| NUMBER OF SENDS TO THIS PORT                  |6
  |-----------------------------------------------|
 7| SOFT INTERRUPT PLABEL                         |7
  |-----------------------------------------------|
```

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

        0 - Both priv and user mode code can be interrupted.

        1 - Only user mode can be interrupted.

Subtype Soft interrupt subtype
         1 - Message file software interrupts.

Message Queue Entry (MQE)

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

Timer entry definitions - 0 - no timeout
                          1 - timeout expired
                          2 - TLE address for a pending timeout

File System Message Files

Wait Message:

param#
  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
|------------------------------------------------|
```

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 number	MQE address	Return port

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 X71) which supports a set of message subqueues which are private to that process. There are a maximum of five 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 awakened from a message wait.

A 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 6 words in length with 4 words available for data. 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 C-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.

```
| 0| 1| 2| 3| 4| 5| 6|
|--|--|--|--|--|--|--|
|LS| L|    DATA     |
|--|--|-------------|
```

LS = Subqueue or Link
L = Length (2-6)

Supporting Data Structures

Message Harbor Table

DST = X71 (57)

```
     |---------------------------------|
   0 | DST INDEX NUMBER  (X71)         |
     |---------------------------------|
   1 | DATA SEGMENT SIZE               |
     |---------------------------------|
   2 | USER REGION POINTER             |
     |---------------------------------|
   3 | MAXIMUM NUMBER OF PINS + 1      |
     |---------------------------------|
   4 | MAXIMUM MSG SIZE (6)            |
     |---------------------------------|
   5 | MAX CONTEXT SIZE                |
     |---------------------------------|
   6 | MESSAGE POOL HEAD POINTER       |
     |---------------------------------|
   7 | MESSAGE POOL TAIL POINTER       |
     |---------------------------------|
 | 10| AVAILABLE MSG FRAMES COUNT      |
     |---------------------------------|
 | 11| HEAD OF IMPEDED QUEUE           |
     |---------------------------------|
 | 12| TAIL OF IMPEDED QUEUE           |
     |---------------------------------|
 | 13| MAX # OF PENDING MSGS           |
     |---------------------------------|
 | 14| CURRENT # OF PENDING MSGS       |
     |---------------------------------|
 | 15| PORTS    (16 WORDS EACH)        |
     |    (6 FOR HEADER + 2 LINK WORDS |
     |    FOR EACH OF 5 SUBQUEUES)     |
     |---------------------------------|
     | MESSAGES  (6 WORDS EACH)        |
     |    (2 FOR HEADER + 4 FOR DATA)  |
     |---------------------------------|
```

** Note: The Message Harbor Table serves as the System Port Data Segment.
The Ports Facility also allows the creation of additional Port Data Segments
which have a similar format. In the header of a Port Data Segment other than
the Message Harbor Table, words X13 and X14 will contain the Timer Head and
the Timer TRLX respectively. Networking software is the primary user of Port
Data Segments although they also use subqueues in the System Port Data
Segment (Message Harbor Table).

Message Harbor Table (Cont.)

Port

```
     |----------------------------|
   0 | FLAGS                      |0
     |----------------------------|
   1 | MASK                       |1
     |----------------------------|
   2 | PIN                        |2
     |----------------------------|
   3 | CONTEXT                    |3
     |----------------------------|
   4 | TYPE                       |4
     |----------------------------|
   5 | PLABEL                     |5
     |----------------------------|
   6 | SUBQ 0 HEAD                |6
     |----------------------------|
   7 | SUBQ 0 TAIL                |7
     |----------------------------|
 | 10| SUBQ 1 HEAD                |8
     |----------------------------|
 | 11| SUBQ 1 TAIL                |9
     |----------------------------|
 | 12| SUBQ 2 HEAD                |10
     |----------------------------|
 | 13| SUBQ 2 TAIL                |11
     |----------------------------|
 | 14| SUBQ 3 HEAD                |12
     |----------------------------|
 | 15| SUBQ 3 TAIL                |13
     |----------------------------|
 | 16| SUBQ 4 HEAD                |14
     |----------------------------|
 | 17| SUBQ 4 TAIL                |15
     |----------------------------|
```

Message Harbor Table (Cont.)

Port/Subqueue Explanations

SUBQUEUE USERS:

```
   Subqueue 0 - Various system process functions.  Examples:
                  PROGEN - DFS Errors
                  IOMESSPROC - Misc. Message Handling
                  SECURITYPROC - Messages for DOWNed Devices
   Subqueue 1 - User Soft Interrupts
   Subqueue 2 - System Soft Interrupts (not used)
   Subqueue 3 - Logon Synch between DO'START & INITJSMP.  Also used
                  by Networking Software.
   Subqueue 4 - LOAD/LOADER1 Communication   .
```

Each process has a port # equaling its pin #. Port 0 is the Kernel Port
used by the Dispatcher & Memory Manager routines. The subqueue assignments
for Port 0 are as follows: .

```
   Subqueue 0 - Make Absent Port
   Subqueue 1 - Segment Modification Complete Port
   Subqueue 2 - Release Region Port
   Subqueue 3 - Fetch Segment Port for I/O Device Monitors
   Subqueue 4 - Cache Move Request Port
```

Port 4 is usually assigned to SYSPORTSERVER (pin 4) which has its own
uses for the subqueues as follows:

```
   Subqueue 0 - Not used
   Subqueue 1 - Port Segment Completor
   Subqueue 2 - Port Timeout
   Subqueue 3 - Not used
   Subqueue 4 - Port Enable
```

Message Harbor Table (Cont.)

Message

```
     |----------------------------|
   0 | LINK                       |0
     |----------------------------|
   1 | LENGTH                     |1
     |----------------------------|
   2 |                            |2
   3 | DATA                       |3
   4 |                            |4
   5 |                            |5
     |----------------------------|
   6 |                            |6
   7 |                            |7
 | 10|                            |8
 | 11|                            |9
     |----------------------------|
```

Timer

```
     |----------------------------|
   0 | LINK                       |
     |----------------------------|
   1 | LENGTH                     |
     |----------------------------|
   2 | REQ ID                     |
     |----------------------------|
   3 | SUB QUEUE                  |
     |----------------------------|
   4 | DELTA TIME                 |
     |                            |
   5 |                            |
     |----------------------------|
   6 | REPLY PORT                 |
     |                            |
   7 |                            |
     |----------------------------|
```

CHAPTER 20 MMSTATS EVENTS

MMSTATS Catalog Index

Event Name	Event No. DEC.	Event No. Z		Event Name	Event No. DEC.	Event No. Z
ALCSTBLK	20	024 (-)	*	FREAD	62	076 (-)
ALLOCMEM	12	014	*	FREADDIR	64	100 (-)
AWAKEDEV	82	122	*			
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
CREAD	160	240	*	MAKEOC	1	001
			*	MAP_DOM	87	127
CSDRIVER	150	226	*			
CSIOWAIT	144	220	*	MONOFF	229	345 (-)
			*	PFAIL	240	360 (-)
CWRITE	149	225	*	PROCESS COMPLETE	211	323 (-)
DC1DC2ACK	231	347 (-)	*	QONOBJ	0	000
			*	QUE_LDR	16	020
DEALLOCM	13	015	*	QUIESCE	40	050
DEALCSTBLK	21	025 (-)	*	RELRESOURCES	23	027 (-)
			*	REQCACHE	90	132
DISKBUGCATCHER	200	310	*	SEGIO	5	005
			*	SIODM-ENTRY	194	302
DISKBUGCATCHER	201	311	*	SIODM-EXIT	195	303
DISKERROR	100	144 (-)	*	SIODONE	6	006
			*	SIOP-WST	193	301
			*	SOFT'DEATH	120	170

MMSTATS Catalog Index (Cont.)

Event Name				Event Name		
DISKERROR	101	145 (-)	*	SPECCHAR	236	354 (-)
DISKINTRPT	191	277	*	SPECIALRQ	2	002
			*	SPECREAD	238	356 (-)
			*	START I/O	110	156 (-)
			*	STACK OVERFLOW	25	31
			*	STRATEGY	83	123
DISK TRAFFIC	98	142 (-)	*	SWAPIN	8	010
DQUE_LDR	17	021	*			
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'	61	075 (-)	*	TERMWRITE	232	350 (-)
			*	TIMESTAMP	228	344 (-)
FPOINT	70	106 (-)	*	UN_MAP_RG	88	130

(-) = Events are not logged in Monitor Table but may be logged on
magnetic tape.

MMSTAT CATALOG INDEX

MMSTAT Event Group 0 (Memory Manager)

Event 0

EVENT NAME:	QONOBJ
DESCRIPTION:	ABSENCE TRAP ON CODE/DATA SEGMENT
CALLING MODULE:	KERNELC
CALLING PROCEDURE(S):	QUEUEONOBJECT

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 Word (From Stack Marker) Of Calling (Trapping) Segment

P5 = P REG Word (From Stack Marker) Of Calling (Trapping) Segment

P6 = Not Used

Event 1

EVENT NAME: MAKEOC

DESCRIPTION: MAKE SEGMENT AN OVERLAY CANDIDATE - RELEASE
                       SEGMENT TO THE POOL OF AVAILABLE SPACE

CALLING MODULE: KERNELC

CALLING PROCEDURE(S): 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 - Not Used

Event 2

EVENT NAME: SPECIALRQ

DESCRIPTION: REQUEST OF SEGMENT EXPANSION/CONTRACTION, UNLOCK,
                       UNFREEZE, IOUNFREEZE, LOCK, IOFREEZE, FREEZE

CALLING MODULE: KERNELC, KERNELD, ININ

CALLING PROCEDURE(S): UNLOCKSEG', IOFREEZE', FETCHOBJECT-(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 (Program Segment Only)

    P2 = Segment Number

P3 = .(0:1) = 1 = If FETCHOBJECT Is Called With I/O FREEZE, FREEZE
              Or LOCK Options
    .(12:4) Type Of Request
          = 0 = IOFREEZE
          = 1 = FREEZE
          = 2 = LOCK
          = 3 = IOUNFREEZE
          = 4 = UNFREEZE
          = 5 = UNLOCK
          = 6 = DLSIZE Expansion
          = 7 = DLSIZE Contraction
          = 10 = PXFIXED Expansion
          = 11 = PXFILE Expansion
          = 12 = PXFILE Contraction
          = 13 = XDS Expansion
          = 14 = XDS Contraction
          = 15 = ZSIZE Expansion
          = 16 = ZSIZE Contraction
          = 17 = 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
    = 6-16 = Requested Size Of Area In Words
    = 17 = S Reg Value When Stack Overflow Occurred

P5 = Status Word If Request Type Is STACKOVERFLOW

P6 = PDISABLE Count If Request Type Is STACKOVERFLOW

Event 4

EVENT NAME: FETCHSEG

DESCRIPTION: SEGMENT REQUEST (FOR I/O SYSTEM OR PROCESS)

CALLING MODULE: KERNELC

CALLING PROCEDURE(S): 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 - Not Used

Event 5

EVENT NAME: SEGIO

DESCRIPTION: MEMORY MANAGEMENT READ/WRITE OF SEGMENT FROM/TO
                       DISC QUEUED

CALLING MODULE: KERNELC

CALLING PROCEDURE(S): 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 - Not Used

Event 6

EVENT NAME: SIODONE

DESCRIPTION: MEMORY MANAGEMENT SEGMENT READ/WRITE FROM/TO DISC
                         COMPLETE

CALLING MODULE: KERNELC

CALLING PROCEDURE(S):   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 - Not Used

Event 7

EVENT NAME: CGARBAGE

EVENT DESCRIPTION: GARBAGE COLLECTION HAS JUST TAKEN PLACE

CALLING MODULE: KERNELC

CALLING PROCEDURE(S):   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-P6 - Not Used

Event 8 (X10)

EVENT NAME: SWAPIN

DESCRIPTION: SWAP IN A PROCESS

CALLING MODULE: KERNELC

CALLING PROCEDURE(S):   SWAPIN

Parameter Description

P1 = PIN of Process Being SWAPPED In

P2 = .(0:1) = 0 = Being SWAPPED
           = 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-P6 - Not Used

MMSTAT Event Group 1 (Memory Manager/Caching)

Event 12 (X14)

EVENT NAME: ALLOCMEM

DESCRIPTION: FOUND A HOLE FOR A SEGMENT REPLACEMENT REQUEST

CALLING MODULE: KERNELC

CALLING PROCEDURE(S):   RESERVEREGION

Parameter Description

P1 = Requested Size In Pages

P2 = Bank Of Selected Region

P3 = Address Of Selected Region

P4-P6 - Not Used

Event 13 (X15)

EVENT NAME: DEALLOCM

DESCRIPTION: RELEASE REGION OF MEMORY TO AVAILABLE STATUS

CALLING MODULE: KERNELC

CALLING PROCEDURE(S):   RELEASEREGION

Parameter Description

P1 = Size Released In Pages

P2 = Bank Of Released Region Base

P3 = Address Of Released Region Base

P4-P6 - Not Used

Event 14 (X16)

EVENT NAME: CACHEMOV

DESCRIPTION: A CACHE MOVE (I.E., LOGICAL DISC REQUEST) HAS
                         JUST COMPLETED

CALLING MODULE: CACHESEG

CALLING PROCEDURE(S):   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 (X17)

EVENT NAME: GET_CDT

DESCRIPTION: CALLED WHEN AN ENTRY IN THE CDT TABLE IS
                         OBTAINED OR RELEASED.

CALLING MODULE: CACHESEG

CALLING PROCEDURE(S):   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-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(S):   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-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(S):   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-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(S):   CDT'FIND'DE

Parameter Description

P1 = LDEV Number Of The CDT Device Entry To Be Found.

P2 = CDT Device Entry

P3-P6 - Not Used

Event 19 (%23)

EVENT NAME: LOCKRANG

DESCRIPTION:

CALLING MODULE:

CALLING PROCEDURE(S):

MMSTAT Event Group 2 (Memory Manager)

Event -20 (-%24)

EVENT NAME: ALCSTBLK

DESCRIPTION: REQUEST TO RESERVE A BLOCK OF ENTRIES IN THE CSTX

CALLING MODULE: KERNELD

CALLING PROCEDURE(S):   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-P6 - Not Used

Event -21 (-%25)

EVENT NAME: DEALCSTBLK

DESCRIPTION: INDICATES THAT A CST EXTENSION BLOCK HAS BEEN
                         DEALLOCATED

CALLING MODULE: KERNELD

CALLING PROCEDURE(S):   DEALCSTBLOCK

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 = NCNT = (NAllocated CSTX Entries- NEntries Being Released)*4

P4-P6 - Not Used

Event -23 (-%27)

EVENT NAME: RELRESOURCES

DESCRIPTION: RESOURCES (VDS,MAIN MEMORY, ST ENTRY) RESERVED
                         FOR THE SEGMENT HAVE BEEN RELEASED

CALLING MODULE: KERNELD

CALLING PROCEDURE(S):   RELDATASEG

Parameter Description

P1 = New DB DST Number

P2 = DELTA P At EXCHANGEDB Call

P3 = Status At EXCHANGEDB Call

P4-P6 - Not Used

Event 25 (%31)

EVENT NAME: STACKOVERFLOW

DESCRIPTION: INDICATES THAT S>2 (NORMAL STACK EXPANSION
                         NEEDED) OR THAT S>MAXDATA (STACK OVERFLOW ABORT)

CALLING MODULE: ININ

CALLING PROCEDURE: STACKOVERFLOW

Parameter Description

P1 = Current process' PCB RESABORTINFO WORD

P2 = Current process' PCB PROCSTATE WORD

P3 = Current process' S Register value

P4 = P Reg within module receiving overflow

P5 = STATUS WORD of module receiving overflow

P6 = PDISABLE count

MMSTAT Event Group 3

(NOT CURRENTLY ASSIGNED)

MMSTAT Event Group 4 (Scheduling)

Event 40 (Z50)

EVENT NAME:	QUIESCE
DESCRIPTION:	PROCESS SWITCH - STATE OF PROCESS SAVED
CALLING MODULE:	KERNELC
CALLING PROCEDURE(S):	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 Inpeded 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-P6 - Not Used

MMSTAT Event Group 5 (IPC/MSG File)

Event -50 (-Z62)

EVENT NAME:	FCPORTOPEN
DESCRIPTION:	OPEN BIPC PORT
CALLING MODULE:	BIPC
CALLING PROCEDURE:	FCPORTOPEN

Parameter Description

P1 = Port Number

P2 = Port DST Number

P3 = Flags

P4-P6 - Not Used

Event -51 (-Z63)

EVENT NAME:	FCPORTRECEIVE
DESCRIPTION:	RECEIVE MESSAGE FROM BIPC PORT
CALLING MODULE:	BIPC
CALLING PROCEDURE:	FCPORTRECEIVE

Parameter Description

P1 = Port Number

P2 = Message Address/Flags

P3 = Return Port

P4-P6 - Not Used

Event -52 (-Z64)

EVENT NAME:	FCPORTSEND
DESCRIPTION:	SEND TO BIPC PORT
CALLING MODULE:	BIPC
CALLING PROCEDURE:	FCPORTSEND

Parameter Description

P1 = Port Number

P2 = Message Address/Flags

P3 = Return Port

P4-P6 - Not Used

Event -53 (-Z65)

EVENT NAME:	PORT STATUS CHANGE
DESCRIPTION:	ENABLE/DISABLE BIPC PORT
CALLING MODULE:	BIPC
CALLING PROCEDURE:	FCPORTENABLE/FCPORTDISABLE

Parameter Description

P1 = Port Number

P2 = 0 = Enable; 1 = Disable

P3 = Address of First Message

P4-P6 - Not Used

Event -54 (-Z66)

EVENT NAME:	FCMSGABORT
DESCRIPTION:	PURGE MESSAGES
CALLING MODULE:	BIPC
CALLING PROCEDURE:	FCMSGABORT

Parameter Description

P1 = Port Number

P2 = Match Parameter

P3 = Return Port

P4-P6 - Not Used

Event -55 (-Z67)

EVENT NAME:	FCPORTCLOSE
DESCRIPTION:	CLOSE BIPC PORT
CALLING MODULE:	BIPC
CALLING PROCEDURE:	FCPORTCLOSE

Parameter Description

P1 = Port Number

P2 = Port DST Number

P3 = Number of Ports Left Open

P4-P6 - Not Used

Event -56 (-Z70)

EVENT NAME:	EXPANDPORT SEG
DESCRIPTION:	EXPAND BIPC PORT TABLE
CALLING MODULE:	BIPC
CALLING PROCEDURE:	EXPANDPORTSEG

Parameter Description

P1 = Port DST Number

P2 = Number of Blocks Added

P3 = Total Number of Blocks

P4-P6 - Not Used

Event -57 (-X71)

EVENT NAME:	TIMEOUT EXPIRED
DESCRIPTION:	MESSAGE TIMER EXPIRED
CALLING MODULE:	FCPOSTIMEOUT
CALLING PROCEDURE:	FCPOSTIMEOUT

Parameter Description

P1 = Port Number

P2 = Message Address

P3 = Return Port

P4-P6 - Not Used

Event -58 (-X72)

EVENT NAME:	IPC INTERNAL EVENT
DESCRIPTION:	IPC INTERNAL EVENT
CALLING MODULE:	IPC
CALLING PROCEDURE:	MAKEMMSTAT

Parameter Description

The parameter values are a function of the event and the first four
(4) bits of parameter 1, which is a subtype.

P1 Bits (0:4) - Subtype
(4:2) - File State
0 = Empty
1 = Non-empty
2 = Less Than One Full Record Left
3 = Full
(6:1) = 1 = Waiting Readers
(7:1) = 1 = Waiting Writers
(11:1) = Carriage Control Characters
(12:4) = Local Flags

Event/ Subtype	Name	P2 (0:8)	P2 (8:8)	P3
72/0	Read Initiation	Record Number	ID	Number of Records
72/1	Read Completion	Error	ID	Number of Records
72/2	Write Initiation	Record Number	ID	Free Records
72/3	Write Completion	Error	ID	Free Records
72/4	Control	Error	ID	(0:4) = Function
				(4:12) = Parameter
72/5	EOF	Error	ID	Number of Records
72/6	Open	Error	ID	Number of Records
72/7	Close	Free Records	ID	Number of Records
72/10	Initialization	0	0	(0:8) = FN
				(8:8) = Update

P4-P6 - Not Used

Event -59 (-X73)

EVENT NAME:	IPC INTERNAL EVENT
DESCRIPTION	IPC INTERNAL EVENT
CALLING MODULE:	IPC
CALLING PROCEDURE:	MAKEMMSTAT

Parameter Description

The parameter values are a function of the event and the first four
(4) bits of parameter 1, which is a subtype.

P1 Bits (0:4) = Subtype
(4:2) = File State
0 = Empty
1 = Non-empty
2 = Less Than One Full Record Left
3 = Full
(6:1) = 1 = Waiting Readers
(7:1) = 1 = Waiting Writers
(11:1) = Carriage Control Characters
(12:4) = Local Flags

Event/ Subtype	Name	P2 (0:8)	P2 (8:8)	P3
73/0	Put Record	Error	ID	(0:3) = Rectype (3:12) = # of Records
73/1	Delete Record	Error	ID	(0:3) = Rectype (3:12) = # of Records
73/2	Delete Block	Start of File Block Number		End of File Block

P4-P6 - Not Used

MMSTAT Event Group 6 (FILESYS)

These events are for development use only and are not normally enabled.

Event -60 (-X74)

EVENT NAME:	FOPEN
DESCRIPTION:	OLD FILE OPEN
CALLING MODULE:	FILEACC
CALLING PROCEDURE(S):	FOPENOA

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 (-X75)

EVENT NAME: FOPEN'

DESCRIPTION: OLD FILE OPEN (CONTINUATION OF EVENT -60)

CALLING MODULE: FILEACC

CALLING PROCEDURE(S):   FOPENDA

Parameter Description

P1 = File Label File Limit - MSW

P2 = File Label File Limit - LSW

P3 = File Label # Of Extents

P4-P6 - Unused

Event -60 (-X74)

EVENT NAME: FOPEN

DESCRIPTION: NEW DISC FILE OPEN

CALLING MODULE: FILEACC

CALLING PROCEDURE(S):   FOPEN

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 (-X75)

EVENT NAME: FOPEN'

DESCRIPTION: NEW DISC FILE OPEN (CONTINUATION OF EVENT -60)

CALLING MODULE: FILEACC

CALLING PROCEDURE(S):   FOPEN

Parameter Description

P1 = FCB File Limit

P2 = FCB Max # Extents

P3 = (0:8)= Initial Allocation Extents

P4-P6 - Not Used

Event -62 (-X76)

EVENT NAME: FREAD

DESCRIPTION:

CALLING MODULE: FILEIO

CALLING PROCEDURE(S):   FREAD

Parameter Description

P1 = File #

P2 = ACBTLOG - Transfer Count

P3 = FLAGS - (15:1) Buffer Hit Flag

Event -63 (-X77)

EVENT NAME: FWRITE

DESCRIPTION:

CALLING MODULE: FILEIO

CALLING PROCEDURE(S):   FWRITE

Parameter Description

P1 = File #

P2 = TCOUNT - See Intrinsics Manual

P3 = FLAGS - (15:1) Buffer Hit Flag

Event -64 (-X100)

EVENT NAME: FREADDIR

DESCRIPTION:

CALLING MODULE: FILEIO

CALLING PROCEDURE(S):   FREADDIR

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 (-X101)

EVENT NAME: FWRITEDIR

DESCRIPTION:

CALLING MODULE: FILEIO

CALLING PROCEDURE: FWRITEDIR

Parameter Description

P1 = File #

P2 = TCOUNT - See Intrinsics 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(S): FUPDATE

Parameter Description

P1 = File #
P2 = TCOUNT - See Intrinsics Manual
P3 = FLAGS - (15:1) Buffer Hit Flag
P4-P6 - Not Used

Event -67 (-%103)

EVENT NAME: IOWAIT
DESCRIPTION:
CALLING MODULE: FILEIO
CALLING PROCEDURE(S): IOWAIT

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(S): FREADSEEK

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(S): FSPACE

Parameter Description

P1 = File #

P2 = DISPLACEMENT - See Intrinsics 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(S): FPOINT

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(S): FCONTROL

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(S): FSETMODE

Parameter Description

P1 = File #
P2 = MODEFLAGS - See Intrinsics Manual
P3-P6 - Not Used

Event -74 (-%112)

EVENT NAME: FCHECK
DESCRIPTION:
CALLING MODULE: FILEIO
CALLING PROCEDURE(S): FCHECK

Parameter Description

P1 = File #
P2 = ERRORCODE - See Intrinsics Manual
P3-P6 - Not Used

Event -75 (-X113)

EVENT NAME: FGETINFO

DESCRIPTION:

CALLING MODULE: FILEIO

CALLING PROCEDURE(S): FGETINFO

Parameter Description

P1 = File #

P2 = FOPTIONS - See Intrinsics Manual

P3 = AOPTIONS - See Intrinsics Manual

P4-P6 - Not Used

Event -76 (-X114)

EVENT NAME: FREADLABEL

DESCRIPTION:

CALLING MODULE: FILEIO

CALLING PROCEDURE(S):

Parameter Description

P1 = File #

P2 = TCOUNT - See Intrinsics Manual

P3-P6 - Not Used

Event -77 (-X115)

EVENT NAME: FWRITELABEL

DESCRIPTION:

CALLING MODULE: FILEIO

CALLING PROCEDURE(S): FWRITELABEL

Parameter Description

P1 = File #

P2 = TCOUNT - See Intrinsics Manual

P3-P6 - Not Used

Event -78 (-X116)

EVENT NAME: FLOCK

DESCRIPTION:

CALLING MODULE: FILEIO

CALLING PROCEDURE(S): FLOCK

Parameter Description

P1 = File #

P2 = LOCKCOND - See Intrinsics Manual

P3 = COND CODE - See Intrinsics Manual

Event -79 (-X117)

EVENT NAME: FUNLOCK

DESCRIPTION:

CALLING MODULE: FILEIO

CALLING PROCEDURE(S): FUNLOCK

Parameter Description

P1 = File #

P2-P6 - Not Used

MMSTAT Event Group 8 (FILESYS/Caching)

Event -80 (-X120)

EVENT NAME: FRENAME

DESCRIPTION:

CALLING MODULE: FILEACC

CALLING PROCEDURE(S): FRENAME

Parameter Description

P1 = File #

P2-P6 - Not Used

Event -81 (-X121)

EVENT NAME: FCLOSE

DESCRIPTION:

CALLING MODULE: FILEACC

CALLING PROCEDURE(S): FCLOSE

Parameter Description

P1 = File #

P2 = DISP - See Intrinsics manual

P3 = SECCODE

P4-P6 - Not Used

Event 82 (X122)

EVENT NAME: AWAKEDEV

DESCRIPTION: AWAKES I/O DEVICE MONITOR WHEN SEGMENT FETCH COMPLETES
CALLING MODULE: KERNELC

CALLING PROCEDURE(S): PROCESSSCHEDMSG,UNDEFEROBJSMPQ

Parameter Description

P1 = SYSDB RELATIVE DIT POINTER OF LDEV TO BE AWAKENED

P2 = WORD 0 (FLAGS WORD) OF THE DIT OF THE LDEV TO BE AWAKENED

P3 = IF DATA OBJECT THEN IOQ OR DRQ INDEX OTHERWISE LDEV NUMBER

P4-P6 - Not Used

Event 83 (Z123)

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(S): 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-P6 - Not Used

Event 84 (Z124)

EVENT NAME: INITIATE

DESCRIPTION: CALLED WHEN STARTING/COMPLETING LOGICAL DISC
                       REQUEST

CALLING MODULE: CACHESEG

CALLING PROCEDURE(S): CDT'INITIATOR, CDT'COMPLETOR

Parameter Description

P1 = CDT Mapped Domain Entry Number

P2 = LDR Entry Index

P3 = Type
     0 = Initiator
     1 - Completor

P4-P6 - Not Used

Event 85 (Z125)

EVENT NAME:

DESCRIPTION:

CALLING MODULE: HARDRES

CALLING PROCEDURE(S): SIODM

Event 86 (Z126)

EVENT NAME: CDT_ATT

DESCRIPTION: CALLED FROM CDT'ATTACHIO

CALLING MODULE: CACHESEG

CALLING PROCEDURE(S): CDT'ATTACHIO

Parameter Description

P1 = Ldev

P2 = Function

P3 = Flags

P4-P5 = Parm1, Parm2

P6 = Count

Event 87 (Z127)

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-P6 - Not Used

Event 88 (Z130)

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-P6 - Not Used

Event 89 (Z131)

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-P6 - Not Used

Event -98 (-%142)

EVENT NAME: DISK TRAFFIC

DESCRIPTION: DISC I/O REQUEST HAS BEEN QUEUED

CALLING MODULE: HARDRES

CALLING PROCEDURE(S): ATTACHIO

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 (Disc Errors)

Event 100 (%144)

EVENT NAME: DISK ERROR

DESCRIPTION: RECORD DISC ERROR

CALLING MODULE: IOFDISC1

CALLING PROCEDURE(S): FHDDVR

Parameter Description

P1 = DIPT(DSTAT) - Hardware Status

P2 = SO - QMISC

P3 = IOQP(QLDEV).QLDEVN LOR STOCOUNT&LSL(8)) = DEV/SIO Program Counter

Event 101 (%145)

EVENT NAME: DISK ERROR

DESCRIPTION: RECORD DISC ERROR

CALLING MODULE: IOMDISCO

CALLING PROCEDURE(S): MHDDVR

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 (SIO)

Event -110 (-%156)

EVENT NAME: START I/O

DESCRIPTION: DRIVER INITIATOR FOR SIO DEVICE HAS BEEN CALLED

CALLING MODULE: HARDRES

CALLING PROCEDURE(S): SIODM

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(QWBCT)=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(S): SIODM

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 (Disc Space)

Event 120 (%170)

EVENT NAME: SOFT'DEATH

DESCRIPTION: BUG CATCHER

CALLING MODULE: HARDRES

CALLING PROCEDURE(S): SOFT'DEATH

Parameter Description

P1 = SOFT'DEATH I.D. Number

P2 = Caller's Status Register

P3 = Caller's Delta P

Event 125 (%175)

EVENT NAME: IOBUFTRP

EVENT DESCRIPTION: IOSYSTEM BUFFER TRAP

CALLING MODULE: HARDRES

CALLING PROCEDURE(S): SIODM

Parameter Description

P1 = IOQP

P2 = IOQP(QDSTN).DSTN = DST Number Of Buffer

P3 = 0

Event -130 (-X202)

EVENT NAME:

DESCRIPTION:

CALLING MODULE: HARDRES

CALLING PROCEDURE(S):   ATTACHIO

Parameter Description

P1 = LDEV

P2 = P Register

P3 = RSTATUS

P4-P6 - Not Used

Event -131 (-X203)

EVENT NAME:

DESCRIPTION:

CALLING MODULE: HARDRES

CALLING PROCEDURE(S):   ATTACHIO

Parameter Description

P1, P2 = Extent Base

P3   = Extent Size

P4-P6 - Not Used

Event -132 (-X204)

EVENT NAME:

DESCRIPTION:

CALLING MODULE: HARDRES

CALLING PROCEDURE: ATTACHIO

Parameter Description

P1, P2 = Formal Parameters Given To ATTACHIO Which Are Device
          Dependent Parameters

P3   = Formal.FLAGS Parameter Supplied To ATTACHIO By The Caller

P4-P6 - Not Used

MMSTAT Event Group 13 (Disc Caching)

Event 139 (X213)

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 (X214)

EVENT NAME: COPEN

DESCRIPTION:

CALLING MODULE: COMSYS2

CALLING PROCEDURE(S): COPEN

Parameter Description

P1 = (0:8) = CS Error Code
   = (8:8) = Logical Device Number

P2 = PMAP1

P3 = PMAP2

Event 142 (X216)

EVENT NAME: CABORTIO

DESCRIPTION:

CALLING MODULE: COMSYS1

CALLING PROCEDURE(S): CABORTIO

Parameter Description

P1 = Logical Device Number

P2 = IOGINDEX

P3 = 0

Event 144 (Z220)

EVENT NAME: CSIOWAIT

DESCRIPTION:

CALLING MODULE: CONSYS1

CALLING PROCEDURE(S): CSIOWAIT

Parameter Description

P1 = (0:8) = CS Error Code
   = (8:8) = Logical Device Number

P2 = Transmission Log

P3

Event 146 (Z222)

EVENT NAME: CCLOSE

DESCRIPTION:

CALLING MODULE: CONSYS3

CALLING PROCEDURE(S): CCLOSE

Parameter Description

P1 (0:8) = CS Error Code
    (8:8) = Logical Device Number

P2 = Line Number

P3 = 0

Event 147 (Z223)

EVENT NAME: CREAD

DESCRIPTION:

CALLING MODULE: CONSYS4

CALLING PROCEDURE(S): CREAD

Parameter Description

P1 = (0:8) = CS Error Code
     (8:8) = Logical Device Number

P2 = INCOUNT

P3 = STATION

Event 149 (Z225)

EVENT NAME: CWRITE

DESCRIPTION:

CALLING MODULE: CONSYS4

CALLING PROCEDURE(S): CWRITE

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 (Z226)

EVENT NAME: CSDRIVER

DESCRIPTION:

CALLING MODULE: BSCLCM

CALLING PROCEDURE(S): CSDRIVER

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 That Caused The Driver To
                            Become Active

Event 152 (Z230)

EVENT NAME: CCONTROL

DESCRIPTION

CALLING MODULE: CONSYS5

CALLING PROCEDURE(S): CCONTROL

Parameter Description

P1 = (0:8) = CS Error Code
   = (8:8) = Logical Device Number

P2 = Control Code

P3 = Parameter

Event 153 (Z231)

EVENT NAME: COPENTRACEFILE

DESCRIPTION:

CALLING MODULE:

CALLING PROCEDURE(S): COPENTRACEFILE

Parameter Description

P1 = (0:8) = CS Error Code
   = (8:8) = Logical Device Number

P2 = CTRACEINFO

P3 = 0

Event 154 (Z232)

EVENT NAME: CCLOSETRACEFILE

DESCRIPTION:

CALLING MODULE:

CALLING PROCEDURE(S): CCLOSETRACEFILE

Parameter Description

P1 = (0:8) = CS Error Code
   = (8:8) = Logical Device Number

P2 = 0

P3 = 0

Event 155 (X233)

EVENT NAME: CPOLLIST

DESCRIPTION:

CALLING MODULE:

CALLING PROCEDURE(S): CPOLLIST

Parameter Description

P1 = Logical Device

P2 = CS Error Code

P3 = PMAP

G.23.00
20- 53

MMSTAT Event Group 16 (CS/3000)

Event 160 (X240)

EVENT NAME: CREAD

DESCRIPTION:

CALLING MODULE: DSMON

CALLING PROCEDURE(S):

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		

G.23.00
20- 54

MMSTAT Event Group 19 (Disc Controller Intrpt)

Event 191 (X277)

EVENT NAME: DISKINTRPT

DESCRIPTION: A 7905/7920 CONTROLLER IS PROCESSING AN ATTENTION
                     INTERRUPT (ONLINE/OFFLINE)

CALLING MODULE: HARDRES

CALLING PROCEDURE(S): SIODM

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 X300 and an X303 for each SIO PRGM.
A single isolated (in time) request will generate at least a
X303, X300, X303. If the queue of IOO'S on a DIT never empties,
there would be one X300 and one X303 per SIO PRGM.

G.23.00
20- 55

Event 192 (X300)

EVENT NAME: GIPINTERRUPT

DESCRIPTION: INTERRUPT JUST PROCESSED

CALLING MODULE: HARDRES

CALLING PROCEDURE(S): GIP

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 (X301)

EVENT NAME: STARTIO

DESCRIPTION: ISSUING SIOP MACHINE INSTRUCTION

CALLING MODULE: HARDRES

CALLING PROCEDURE(S): START'HPIB, STARTIO

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

G.23.00
20- 56

Event 194 (Z302)

EVENT NAME: SIODM-ENTRY

DESCRIPTION: ENTERING SIODM

CALLING MODULE: HARDRES

CALLING PROCEDURE(S): SIODM

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 = Not Used

P6 = LSW Returned By Call To TIMER

Event 195 (Z303)

EVENT NAME: SIODM-EXIT

DESCRIPTION: LEAVING SIODM MAIN LOOP

CALLING MODULE: HARDRES

CALLING PROCEDURE(S): SIODM

Parameter Description

The same as Event 194 (Z302), above.

MMSTAT Event Group 20 (Private Volumes)

These Events are for development use only and are not normally enabled.

Event 200 (Z310)

EVENT NAME: DISKBUGCATCHER

DESCRIPTION: A MOUNTED VOLUME TABLE CHANGE IS BEING MADE.

CALLING MODULE: PVSYS

CALLING PROCEDURE(S): MVTABLE

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 (Z311)

EVENT NAME: DISKBUGCATCHER

DESCRIPTION: A PRIVATE VOLUME USER TABLE CHANGE IS BEING MADE.

CALLING MODULE: PVSYS

CALLING PROCEDURE(S): USERTABLE

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 Creation And Termination)

Event -211 (-Z323)

EVENT NAME: PROCESS COMPLETION

DESCRIPTION: PROCESS HAS TERMINATED

CALLING MODULE: MORGUE

CALLING PROCEDURE(S): TERMINATE

Parameter Description

P1 = 0

P2 = 0

P3 = 0

MMSTAT Event Group 22 (Monitor Config Information)

Event 221 (Z335)

EVENT NAME: CONFIGURATION INFORMATION

DESCRIPTION: EVENT GROUP MASK

CALLING MODULE: CRIO

CALLING PROCEDURE(S): CONSMON

Parameter Description

P1 = MEASMSK0

P2 = MEASMSK1

P3 = Reserved

Event 222 (Z336)

EVENT NAME: CONFIGURATION INFORMATION

DESCRIPTION: MPE VERSION FIX UPDATE

CALLING MODULE: OPCOMMAND

CALLING PROCEDURE(S):   CXMON

Parameter Description

P1 = Version

P2 = FIXL

P3 = UPDATEL

Event -223 (-X337)

EVENT NAME: CONFIGURATION INFORMATION

DESCRIPTION: SYSTEM TABLE LOCATIONS AND AVAILABLE LINKED MEMORY
                        INFORMATION

CALLING MODULE: OPCOMMAND

CALLING PROCEDURE(S):   CXMON

Parameter Description

P1 = F (X1032)=@CST(0)-@DST(0) =Displacement To Code
P2 = F(X1033)=@CST(LAST)-@DST(0) =Displacement To Sharable
P3 = LOGICAL(TOTAL&DLSK(4))=Linked Memory Size

Event -224 (-X340)

EVENT NAME: SYSPINS

DESCRIPTION: LOGICAL PROCESS TABLE

CALLING MODULE: OPCOMMAND

CALLING PROCEDURE(S):   CXMON

Parameter Description

P1 = ABSOLUTE(X1141)=PROGEN'S PCB Entry Number

P2 = ABSOLUTE(X1142)=MAM'S PCB Entry Number

P3 = ABSOLUTE(X1143)=UCOP'S PCB Entry Number

Event -225 (-X341)

EVENT NAME: SYSPINS(CNTD.)

DESCRIPTION: LOGICAL PROCESS TABLE

CALLING MODULE: OPCOMMAND

CALLING PROCEDURE(S):   CXMON

Parameter Description

P1 = ABSOLUTE(X1144)=PFAIL'S PCB Entry Number

P2 = ABSOLUTE(X1145)=DEVREC'S PCB Entry Number

P3 = ABSOLUTE(X1146)=PRMSG'S PCB Entry Number

Event -226 (-X342)

EVENT NAME: SYSPINS(CNTD.)

DESCRIPTION: LOGICAL PROCESS TABLE

CALLING MODULE: OPCOMMAND

CALLING PROCEDURE(S):   CXMON

Parameter Description

P1 = ABSOLUTE(X1147)=STMSG'S PCB Entry Number

P2 = ABSOLUTE(X1150)=LOG'S PCB Entry Number

P3 = ABSOLUTE(X1151)=LOAD'S PCB Entry Number

Event -227 (-X343)

EVENT NAME: SYSPINS(CNTD.)

DESCRIPTION: LOGICAL PROCESS TABLE

CALLING MODULE: OPCOMMAND

CALLING PROCEDURE(S):   CXMON

Parameter Description

P1 = ABSOLUTE(X1152)=IOMESSPROC'S PCB Entry Number

P2 = ABSOLUTE(X1153)=SYSIOPROC'S PCB Entry Number

P3 = ABSOLUTE(X1154)=MEMLOGP'S PCB Entry Number

Event -228 (-X344)

EVENT NAME: TIMESTAMP

DESCRIPTION: TIMESTAMP

CALLING MODULE: OPCOMMAND

CALLING PROCEDURE(S):   CXMON

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 (-X345)

EVENT NAME: MONOFF

DESCRIPTION: END EVENT TRACING

CALLING MODULE: OPCOMMAND

CALLING PROCEDURE(S):   CXMON

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(S): TIP

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(S): TIP

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(S): TERMIOM

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(S): TIP

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(S): TERMIOM

Parameter Description

P1 = LDEV

P2 = 0

P3 = 0

Event 235 (%353)

EVENT NAME: TERMLOGOFF

DESCRIPTION: TERMINAL JUST LOGGED OFF

CALLING MODULE: IOTERMO

CALLING PROCEDURE(S): TERMIOM

Parameter Description

P1 = LDEV

P2 = 0

P3 = 0

Event 236 (%354)

EVENT NAME: SPECCHAR

DESCRIPTION: PROCESSED SPECIAL CHARACTER

CALLING MODULE: HARDRES

CALLING PROCEDURE(S): TIP

Parameter Description

P1 = LDEV

P2 = Special Character Processed

P3 = 0

Event 237 (%355)

EVENT NAME: BREAK

DESCRIPTION: PROCESSED BREAK

CALLING MODULE: HARDRES

CALLING PROCEDURE(S): TIP

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(S): TIP

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 PROCEDURE(S): 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

P = 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

Event -241 (-%361)

EVENT NAME: PSEUDOINT

DESCRIPTION: PSEUDO/SOFT INTERRUPT HANDLING

CALLING MODULE: MISCSEGC

CALLING PROCEDURE(S): PSUEDOINT

Parameter Description

P1 = Interrupt Type
     = 0 If Hard Kill (P2, P3 Not Used)
     = 1 If Soft Kill (P2, P3 Not Used)
     = 2 If Control-Y (P2, P3 Not Used)
     = 3 If Break (P2, P3 Not Used)
     = 4 If System Soft Interrupt
         P2 =
         P3 =
     = 5 If User Soft Interrupt
         P2 =
         P3 =

P2, P3 = Dependent on P1

P4, P5, P6 - Not Used

CHAPTER 21   ROOTFILE LAYOUT

General Rootfile Layout

```
            |------------------------|
LABEL 0| ROOTFILE INFORMATION   |
       |    (128 WORDS)         |
       |------------------------|
       |                        |
     1 | PASSWORD TABLE         |
       |                        |
       |------------------------|
     2 | PASSWORD TABLE (CONT.) |
       |------------------------|
       |                        |
     3 | ITEM R/W TABLE         |
       |------------------------|
       .                        .
       .                        .
       .                        .
       .  |------------------------|
       .  | SET R/W TABLE          |
       .  |------------------------|
RECORD 0| DATABASE GLOBAL INFO   |
       |    (128 WORDS)         |
       |------------------------|
     1 | ITEM MAP               |
       | SET MAP                |
       |------------------------|
     2 | ITEM TABLE             |
     . | (VARIABLE SIZE)        |
     . |------------------------|
     . | SET TABLE              |
     . | (VARIABLE SIZE)        |
     . |------------------------|
     . | DATA SET CONTROL BLOCKS|
     . |                        |
     . |     (DSCB)             |
     . |                        |
     . |    (VARIABLE SIZE)     |
     . |------------------------|
     . | DEVICE CLASS TABLE     |
     . | (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

```
    |------------------------------------------|
  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'DBG'NUM  - (DBG DST NUMBER)           | 7
    |------------------------------------------|
 10 | RL'LOGID   -    (LOG ID FOR              | 8
 11 |                 TRANSACTION LOGGING)     | 9
 12 |                                          | 10
 13 |                                          | 11
    |------------------------------------------|
 14 | RL'LOGPASS -    (LOG ID PASSWORD)        | 12
 15 |                                          | 13
 16 |                                          | 14
 17 |                                          | 15
    |------------------------------------------|
 20 | RL'FLAGS   -    (DATABASE FLAGS)         | 16
    |------------------------------------------|
 21 | RL'STORDATE -   (DBSTORE DATE)           | 17
    |------------------------------------------|
 22 | RL'STORTIME     (DBSTORE TIME)           | 18
 23 |                                          | 19
    |------------------------------------------|
 24 | RL'BUFSPECCOUNT (BUFFER SPEC COUNT)      | 20
    |------------------------------------------|
```

Root File Label 0 (Cont.)

```
    |------------------------------------------|
 25 | RL'ILRCREATEDATE (DATE ILR LOG CREATED)  | 21
    |------------------------------------------|
 26 | RL'ILRCREATETIME (TIME ILR LOG CREATED)  | 22
 27 |                                          | 23
    |------------------------------------------|
 30 | RL'ILRLASTDATE   (LAST LOG ACCESS DATE)  | 24
    |------------------------------------------|
 31 | RL'ILRLASTTIME   (LAST LOG ACCESS TIME)  | 25
 32 |                                          | 26
    |------------------------------------------|
 33 | RL'RBPREDATE   (PREVIOUS ROLLBACK DATE)  | 27
 34 | RL'RBPRETIME   (PREVIOUS ROLLBACK TIME)  | 28
 35 |                                          | 29
    |------------------------------------------|
 36 | RL'RBDATE        (ROLLBACK DATE)         | 30
    |------------------------------------------|
 37 | RL'RBTIME        (ROLLBACK TIME)         | 31
 40 |                                          | 32
    |------------------------------------------|
 41 | RESERVED                                 | 33
    |------------------------------------------|
 42 | RL'LANGUAGE'ID    (LANGUAGE ID)          | 34
    |------------------------------------------|
 43 | RL'LANG'MNEMONIC  (LANGUAGE MNEMONIC)    | 35
  - |                                          |
 52 |                                          | 42
    |------------------------------------------|
 53 | RESERVED FOR DBCONV                      | 43
    |------------------------------------------|
 54 | RESERVED                                 | 44
  - |         FOR                              |
  - |              FUTURE                      |
 77 |                   USE                    | 63
    |------------------------------------------|
100 | RL'MAINTWORD    (DATABASE MAINTENANCE    | 64
101 |                  WORD)                   | 65
102 |                                          | 66
103 |                                          | 67
    |------------------------------------------|
104 | RL'BUFFERSPECS   (BUFFER SPECIFICATIONS) | 68
  - |                                          | -
177 |                                          | 127
    |------------------------------------------|
```

Root File Label 0 (Cont.)

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.
    IE - ILR enable in progress.
    ID - ILR disable in progress.
    CN - Conversion by DBCONV was in progress and cannot be continued.
    CA - Conversion by DBCONV was in progress and can be continued.
    MV - Database file move is in progress.

RL'DATE - Root file creation date*. The format is:

```
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
|    YEAR         |    DAY OF YEAR              |
|-----------------|-----------------------------|
```

RL'TIME - Root file creation time*. The format is:

```
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
| 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) - OUTPUT DEFER   Default is NO (0)
    (5:2) - SUBSYSTEM ACCESS Default is R/W (00)
    (7:1) - ILR   Default is NO (0)
    (8:1) - ROLLBACK   Default is NO (0)
    (9:1) - RESERVED
    (10:1)- DIRTY FLAG   Default is YES (1).
                       This indicates the database has been modified but not DBSTOREd.
    (11:1)- DBRECOV RESTART Default is NO (0)
    (12:4)- RESERVED

Root File Label 0 (Cont.)

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'RBPREDATE - Same format as RL'DATE*.

RL'RBPRETIME - Same format as RL'TIME*.

RL'RBDATE - Same format as RL'DATE*.

RL'RBTIME - Same format as RL'TIME*.

RL'LANGUAGE'ID - Same format as defined in system configuration.

RL'LANG'MNEMONIC - Language mnemonic for this database.
                    Maximum 16 characters.

RL'MAINTWORD - For data bases with no maintenance word this field has 2 semicolons (';;') and trailing blanks.

RL'BUFFSPECS -

```
      0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
      |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
  104 | BUFFERS FOR 1 USER    | BUFFERS FOR 2 USERS   | 68
      |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
  105 | BUFFERS FOR 3 USERS   | BUFFERS FOR 4 USERS   | 69
    . |                       |                       | .
    . |                       |                       | .
  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

```
            |-----------------------------------|Z
   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
            |-----------------------------------|
  |     10| PASSWORD FOR USER CLASS 2          |8
  |     11|                                    |9
  |     12|                                    |10
  |     13|                                    |11
            |-----------------------------------|
            ~                                   ~
            |-----------------------------------|
  |    174| PASSWORD FOR USER CLASS 31         |124
       175|                                    |125
       176|                                    |126
  |    177|                                    |127
            |-----------------------------------|
```

Label 2

```
            |-----------------------------------|Z
   WORD  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
            |-----------------------------------|
  |     10| PASSWORD FOR USER CLASS 34         |8
  |     11|                                    |9
  |     12|                                    |10
  |     13|                                    |11
            |-----------------------------------|
  |         ~                                   ~
  |       |-----------------------------------|
  |    174| PASSWORD FOR USER CLASS 63         |124
  |    175|                                    |125
  |    176|                                    |126
  |    177|                                    |127
            |-----------------------------------|
```

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 pass-
word 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

```
  |       |-----------------------------------|
  |      0| ITEM1 READ/WRITE BIT MAP          |0
         1|                                    |1
         2|                                    |2
         3|                                    |3
         4|                                    |4
         5|                                    |5
         6|                                    |6
         7|                                    |7
            |-----------------------------------|
  |     10| ITEM2 READ/WRITE BIT MAP          |8
  |     11|                                    |9
  |         ~                                   ~
  |     17|                                    |15
            |-----------------------------------|
  |     20| ITEM3 READ/WRITE BIT MAP          |16
  |     21|                                    |17
  |         ~                                   ~
  |    167|                                    |117
            |-----------------------------------|
  |    170| ITEM16 READ/WRITE BIT MAP         |120
  |    171|                                    |121
  |         ~                                   ~
  |    177|                                    |127
            |-----------------------------------|
```

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 1023 data items in the schema, the maxi-
mum size for this table in user labels would be:

   Max-size = [(1023)*8]/128 = 63.93 => 64 labels.

Root File - Next Label(s)

```
  |       |-----------------------------------|
  |      0| SET1 READ/WRITE BIT MAP           |0
         1|                                    |1
         2|                                    |2
         3|                                    |3
         4|                                    |4
         5|                                    |5
         6|                                    |6
         7|                                    |7
            |-----------------------------------|
  |     10| SET2 READ/WRITE BIT MAP           |8
  |     11|                                    |9
  |         ~                                   ~
  |     17|                                    |15
            |-----------------------------------|
  |     20| SET3 READ/WRITE BIT MAP           |16
  |     21|                                    |17
  |         ~                                   ~
  |    167|                                    |117
            |-----------------------------------|
  |    170| SET16 READ/WRITE BIT MAP          |120
  |    171|                                    |121
  |         ~                                   ~
  |    177|                                    |127
            |-----------------------------------|
```

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 on the next
page.

The number of user labels occupied by the Set Read/Write Table depends on the
number of data sets defined in the schema, and is obtained by rounding up-
wards (ceiling) the result of:

   Num-of-labels = [(Num-of-sets)*8]/128

Since there can only be a maximum of 199 data sets defined in the schema the
maximum size for this table in user labels is:

   Max-size = [(199)*8]/128 = 12.44 = 13 labels

Root File - Next Label(s) (Cont.)

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.

1. Four words for read access:

```
 0       15 16      31 32      47 48      63
|----------|----------|----------|----------|
| WORD 1   | WORD 2   | WORD 3   | WORD 4   |
|----------|----------|----------|----------|
```

Four 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
X000016  X020000  X000410  X001300
```

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.

2. 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
X000010  X020000  X000000  X001100
```

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

```
      |------------------------------------------|
   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
  10|                                            |8
      |------------------------------------------|
  11| ROOT'ITEMCT     (NUMBER OF ITEMS)          |9
      |------------------------------------------|
  12| ROOT'SETCT      (NUMBER OF DATA SETS)      |10
      |------------------------------------------|
  13| ROOT'ITEMPTR    (RECORD # OF ITEM TABLE)   |11
      |------------------------------------------|
  14| ROOT'DSETPTR    (RECORD # OF SET TABLE)    |12
      |------------------------------------------|
  15| ROOT'DSCBPTR    (RECORD # OF DSCB'S)       |13
      |------------------------------------------|
  16| ROOT'DEVICEPTR (RECORD # OF DEVICE CLASS TABLE)|14
      |------------------------------------------|
  17| ROOT'DBGFLAG                               |15
      |------------------------------------------|
  20| RESERVED        (SET TO BLANKS)            |16
  21|                                            |17
  22|                                            |18
  23|                                            |19
      |------------------------------------------|
  24| NOWOPEN                                    |20
      |------------------------------------------|
  25| MAXOPEN                                    |21
      |------------------------------------------|
  26| PR'RESTART'CALENDAR                        |22
      |------------------------------------------|
  27| RR'RESTART'TIMESTAMP                       |23
  30|                                            |24
      |------------------------------------------|
  31| RR'RESTART'FNAME                           |25
  32|                                            |26
  33|                                            |27
  34|                                            |28
      |------------------------------------------|
```

Root File Record 0 (Cont.)

```
      |------------------------------------------|
  35| RE'RESTART'FGROUP                          |29
  36|                                            |30
  37|                                            |31
  40|                                            |32
      |------------------------------------------|
  41| RR'RESTART'FACCT                           |33
  42|                                            |34
  43|                                            |35
  44|                                            |36
      |------------------------------------------|
  45| RESERVED                                   |37
   ~|            (SET TO BINARY 0'S)             ~
 177|_____  |127
```

ROOT'DBSTATUS
        (0:8) - IMAGE version ('C' in ASCII)
        (8:8) - Binary 2 (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 & C.

MAXOPEN - Maximum number of data sets that can be opened. This field is not used in IMAGE B & C.

ROOT'DBGFLAG - 1: Information can fit in DBG.
               0: Information can not fit in DBG.

RR'RESTART'FNAME - Restart file name for DBRECOV stop/restart.

Root File Record 1

```
       0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
      |--|--|--|--|--|--|--|--|--|--|--|--|--|--|
   0|                                            |0
   1|                                            |1
   ~|  ITEM MAP                                  ~
  36|                                            |30
  37|                                            |31
      |------------------------------------------|
  40| RESERVED FOR FUTURE USE                    |32
  41|                                            |33
   ~|                                            ~
  76|                                            |62
  77|                                            |63
      |------------------------------------------|
 100|                                            |64
 101|                                            |65
   ~|  SET MAP                                   ~
 136|                                            |94
 137|                                            |95
      |------------------------------------------|
 140| RESERVED FOR FUTURE USE                    |96
 141|                                            |97
   ~|                                            ~
 176|                                            |126
 177|                                            |127
      |------------------------------------------|
```

The Item Map occupies Words 0-31.

The Set Map occupies Words 64-95.

These two maps are used by DBOPEN for faster access to information in the Item Table and Set Table.

Root File Record 2

```
      0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
    |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
  0| ITEM NAME 1                              |0
  1|                                         |1
  2|                                         |2
  3|                                         |3
  4|                                         |4
  5|                                         |5
  6|                                         |6
  7|                                         |7
    |-----------------------------------------|
 10| ITEM NO. OF SYNONYM                      |8
    |-----------------------------------------|
 11| RESERVED 1          | RESERVED 2         |9
    |-----------------------------------------|
 12| ITEM TYPE           | SUBITEM COUNT      |10
    |-----------------------------------------|
 13| SUBITEM LENGTH      | NOT USED           |11
    |-----------------------------------------|
 14| ITEM NAME 2                              |12
 15|                                         |13
 16|                                         |14
 17|                                         |15
 20|                                         |16
 21|                                         |17
 22|                                         |18
 23|                                         |19
    |-----------------------------------------|
 24| ITEM NO. OF SYNONYM                      |20
    |-----------------------------------------|
 25| RESERVED 1          | RESERVED 2         |21
    |-----------------------------------------|
 26| ITEM TYPE           | SUBITEM COUNT      |22
    |-----------------------------------------|
 27| SUBITEM LENGTH      | NOT USED           |23
 30|                                         |24
```

The Item Table starts in record #2.

Each entry is 12 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

Root File Record 2 (Cont.)

```
              ITEM-TYPE
              |
VALUES, 20J2; |  |SUBITEM-LENGTH
              |  |SUBITEM-COUNT
```

The maximum size for this table is 12*1023 = 12276 words

NOTE: 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(s) Set Table

```
      0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
    |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
  0| SET-NAME-1                               |0
  1|                                         |1
  2|                                         |2
  3|                                         |3
  4|                                         |4
  5|                                         |5
  6|                                         |6
  7|                                         |7
    |-----------------------------------------|
 10| SET-NO-OF-SYNONYM    | RESERVED-1        |8
    |-----------------------------------------|
 11| RESERVED-2           | DATA-SET-TYPE     |9
    |-----------------------------------------|
 12| DSCB-POINTER                             |10
 13|                                         |11
    |-----------------------------------------|
 14| SET-NAME-2                               |12
 15|                                         |13
 16|                                         |14
 17|                                         |15
 20|                                         |16
 21|                                         |17
 22|                                         |18
 23|                                         |19
    |-----------------------------------------|
 24| SET-NO-OF-SYNONYM    | RESERVED-1        |20
    |-----------------------------------------|
 25| RESERVED-2           | DATA-SET-TYPE     |21
    |-----------------------------------------|
 26| DSCB-POINTER                             |22
 27|                                         |23
    |-----------------------------------------|
 30|                                         |24
```

Root File - Next Record(s) (Cont.)

Set Table follows the Item Table. Each entry is 12 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 below.

The maximum size for this table is '2*199 = 2388 words.

NOTE: 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.

Root File - Next Record(s) (Cont.)

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+1  | |
|---                              ---     | |
| PATH TABLE (SET 1)                      | |
|       (SEARCH ITEM, SORT ITEM, ETC.)    | |
|                         PATHCOUNT*3     | |
|-----------------------------------------| /
| 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+1  | |
|---                              ---     | |
| PATH TABLE (SET 2)                      | |
|       (SEARCH ITEM, SORT ITEM, ETC.)    | |
|                         PATHCOUNT*3     | |
|-----------------------------------------| /


|-----------------------------------------| 
| 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+1  | |
|---                              ---     | |
| PATH TABLE (LAST SET)                   | |
|       (SEARCH ITEM, SORT ITEM, ETC.)    | |
|                         PATHCOUNT*3     | |
|-----------------------------------------| /
```

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.

Root File - Next Record(s) (Cont.)

Data Set Control Block (Global Area)

```
       0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
      |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
   0| DSCRP          (DATA SET CAPACITY)      |0
   1|                                         |1
      |-----------------------------------------|
   2| DSBLOCKLGTH    (BLOCK LENGTH)           |2
      |-----------------------------------------|
   3| DSMEDIALGTH    (MEDIA RECORD LENGTH)    |3
      |-----------------------------------------|
   4| DSENTRYLGTH    (ENTRY LENGTH)           |4
      |-----------------------------------------|
   5| DSBLOCKFAC        |    DSPATHCT          |5
      |-----------------------------------------|
   6| DSFIELDCT                                |6
      |--|--------------------------------------|
   7| X| DSPRIMKEY                             |7
      |--|--------------------------------------|
  10| DSPATHPTR   (OFFSET TO PATH TABLE)      |8
      |-----------------------------------------|
  11| LOGICAL END OF FILE                     |9
  12|                                         |10
      |-----------------------------------------|
  13| MAX NUM OF RECORDS IN SET               |11
  14|                                         |12
      |-----------------------------------------|
  15| 17 WORDS OF BINARY ZEROS                |13
   ~ |                                         |
  35|                                         |29
      |-----------------------------------------|
```

DSCRP   - 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.

DSPATHCT   - Data set path count. This is the number of paths that are specified for the data set.

Root File - Next Record(s) (Cont.)

DSFIELDCT   - Data set field count. This is the number of fields 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
      |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
   0| ITEM NUM OF 1ST FIELD                   |
      |-----------------------------------------|
   1| ITEM NUM OF 2ND FIELD                   |
      |-----------------------------------------|
   2| ITEM NUM OF 3RD FIELD                   |
      |-----------------------------------------|
   .| ETC.                                    |
      |-----------------------------------------|
```

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.

Root File - Next Record(s) (Cont.)

Data Set Control Block (Record Definition Item Displacement)

```
       0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
      |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
   0| OFFSET TO 1ST FIELD                     |
      |-----------------------------------------|
   1| WORD OFFSET TO 2ND FIELD                |
      |-----------------------------------------|
   2| WORD OFFSET TO 3RD 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 11, since there are 11 words of overhead (6 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.

Root File - Next Record(s) (Cont.)

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|                                         |
        |-----------------------------------------|
      3| 2ND PATH DEFINITION                     |
      4|                                         |
      5|                                         |
        |-----------------------------------------|
      6|                                         |
   ~   |                                         |~
        |-----------------------------------------|
        | LAST PATH DEFINITION                    |
        |                                         |
        |-----------------------------------------|
```

There are 3 words (6 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
    1-2: item number of the search item in the related detail set.
    3-4: item number of the sort item in the related detail set.
      5: set number of the related detail data set
      6: path number of the corresponding path in the related detail data set.

    Detail sets:
    Byte Description
    1-2: field number of the search item.
    3-4: field number of the sort item.
      5: set number of the related master data set
      6: path number of the corresponding path in the related master data set.

Root File - Next Record(s) (Cont.)

Device Class Table

```
|        0 1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
|       |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
|     0| DEV CLASS NAME 1                             |
|     1|                                             |1
|     2|                                             |2
|     3|                                             |3
|      |----------------------------------------------
|     4| DEV CLASS NAME 2                            |4
|     5|                                             |5
|     6|                                             |6
|     7|                                             |7
|      |----------------------------------------------
|    10| DEV CLASS NAME 3                            |8
|    11|                                             |9
|    12|                                             |10
|    13|                                             |11
|      |----------------------------------------------
|    14| DEV CLASS NAME 4                            |12
|    15|                                             |13
|    16|                                             |14
|    17|                                             |15
|      |----------------------------------------------
|    20|                                             |16
```

Device Class Table follows the DSCBs.

Each entry is 4 words long, and contains the device class name which is optionally specified for a data set by the user. For data sets without user specified device class names, the entries will be filled with blanks.

The length of the table depends on the number of data sets defined in the schema. The relative position of a device class entry depends on its relative position in the schema.

The maximum size for this table is 4*199 = 796 words.

User Label 0

```
          |-----------------------------|
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. Record name consists of an MPE file record number (0 byte to 2nd byte) and a slot number (3rd byte), where the MPE file record number is equal to the quotient of the highest entry divided by blocking factor and the slot number is equal to the remainder of the highest entry divided by blocking factor. 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 should point at the highest entry, '75', in the form of record name. If the data set has a blocking factor of 10, the record name should have an MPE record number of 7 and a slot number of 5.

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 2-3 (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.

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 long (128 words). 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 as follows:

```
       |------------------|
WORD 0 | LARGEST SPACE    |
       |------------------|
WORD 1 | STARTING SPACE   |
       |------------------|
WORD 2 | ENDING SPACE     |
       |------------------|
```

The descriptor table looks as follows:

```
|----------|
|          |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. Therefore, 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| DS'LDEV                     |0
|----------------------------------|
|    1| DS'DST                      |1
|----------------------------------|
|    2|                             |2
|     |- DS'DISC'SIZE             - |
|    3|                             |3
|----------------------------------|
|    4| DS'LAST'PAGE'OF'MAP         |4
|----------------------------------|
|    5| DS'LAST'BUFFER'INDEX        |5
|----------------------------------|
|    6|                             |6
|     |- DS'MAP'ADDRESS           - |
|    7|                             |7
|----------------------------------|
|   10| DS'LOCK                     |8
|----------------------------------|
|   11| DS'LOCK'COUNT               |9
|----------------------------------|
|   12| DS'QUEUE'HEAD               |10
|----------------------------------|
|   13| DS'QUEUE'TAIL               |11
|----------------------------------|
|   14| DS'DESCRIPTOR'TABLE         |12
|----------------------------------|
|   15| DS'BUFFER'PAGE'NUMBER       |13
|----------------------------------|
|   16| DS'BUFFER'DIRTY             |14
|----------------------------------|
|   17| DS'BUFFER'AREA              |15
|----------------------------------|
|   20| DS'FIRST'THRESHOLD'PAGE     |16
|----------------------------------|
|   21|                             |17
|     |- DS'SIZE'OF'LAST'ALLOCATION-|
|   22|                             |18
|----------------------------------|
```

Data Segment Structure (Cont.)

```
|----------------------------------|
|   23| DS'LAST'PAGE'ALLOCATED'FROM |19
|----------------------------------|
|   24| DS'NEXT'BUFFER'INDEX        |20
|----------------------------------|
|   25| DS'PAGE'NUMBER              |21
|----------------------------------|
|   26| DS'WORD'NUMBER              |22
|----------------------------------|
|   27| DS'BIT'NUMBER               |23
|----------------------------------|
|   30| DS'PAGE'POINTER             |24
|----------------------------------|
|   31| DS'STARTING'WORD'NUMBER     |25
|----------------------------------|
|   32| DS'STARTING'BIT'NUMBER      |26
|----------------------------------|
|   33|                             |27
|     |- DS'NUMBER'OF'SECTORS     - |
|   34|                             |28
|----------------------------------|
|   35| DS'BIT'COUNT                |29
|----------------------------------|
|   36| DS'ENTRY'TYPE               |30
|----------------------------------|
|   37| DS'BUFFER'INDEX             |31
|----------------------------------|
|   40|                             |32
|     |- DS'DISC'ADDRESS          - |
|   41|                             |33
|----------------------------------|
|   42| DS'ERROR'STATUS             |34
|----------------------------------|
```

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 their relation to one another, for their 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", which 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/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 needing 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 current workload 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 either a pending or currently active I/O 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, or 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 X1273 and the bank and offset are kept in X1274-X1275. 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, "I/O", for a more detailed explanation of both the DIT and the Disc Request Table header. See Chapter 2, "Memory Management Tables", for a description of the Memory Region Header for a disc domain (cached region).

Disc Caching Tables Overview

Cache Directory Table (CDT)
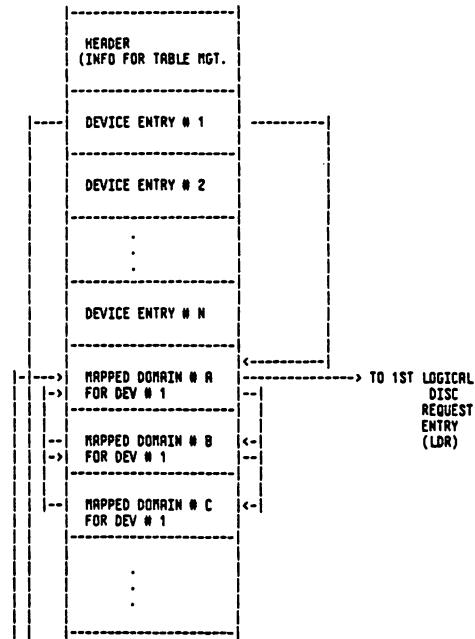
Memory Regions
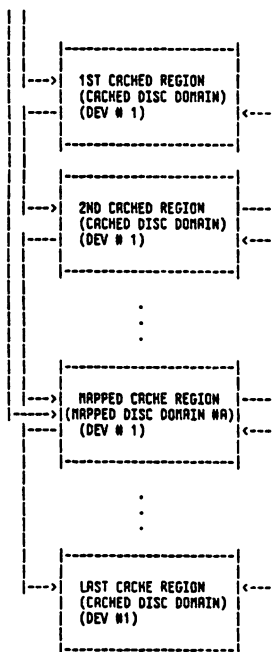
Cache Directory Table

The Cache Directory Table (CDT) is the bookkeeping structure for managing cached disc domains. This table is divided into three parts:

1. CDT Header Entry
   This entry contains all information necessary to manage the entire table and also contains global caching related information.

2. 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.

3. 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 (INI, ROC, etc.) just as the DST Table keeps track of data segments.

The following low core cells contain the address of the CDT:

    X1273 - contains the DST Number of the CDT
    X1274 - contains the Bank Number of the CDT
    X1275 - contains the Offset within the bank of the CDT

Header Entry

```
     |------------------------------------------|
   0 | # ENTRIES                                |CDT'ENTRIES
     |------------------------------------------|
   1 | ENTRY SIZE (Z32)                         |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 |                                          |
     |                .                         |
     |                .                         |
     |                .                         |
  31 |                                          |
     |------------------------------------------|
```

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. (This value may 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.

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, therefore, 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, therefore, 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'SCRNPT
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.

CDT'DE'SHIFT'CNT
The number of bits used to execute DLSL instruction.

CDT'DE'MAKE'EVEN
An additional word used to make the entry size an even number. CDT'ENTRY'SIZE must be an even number for disc caching to correctly access the CDT Device Entry Table and the Mapped Domain Entry Table.

Device Entry

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
30	# BITS TO SHIFT	CDT'DE'SHIFT'CNT
31	NOT USED	CDT'DE'MAKE'EVEN

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 ADDRESS	CDT'MD'SECTOR
4	LAST SECTOR ADDRESS	CDT'MD'END'SECTOR
6	ABSENT/IMI/IMO/MISS/LOCK/FWIP/ROC/VIRGIN/NNPOST/SEQ/STATE	CDT'MD'FLAGS
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 IF PRESENT	CDT'MD'MEM'ADR
16	DRQ FOR THIS MAPPED DOMAIN	CDT'MD'DISCREQ
17	# FLUSHING CDT'S	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		CDT'MD'DEFERRED
	ENTRY LENGTH	
31		

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) - NNPOST.
        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

X1017   Pointer to Logical Disc Request Table

NOTE: This table is really part of the DRQ (refer to Chapter 13). Any entry
      with the logical request bit set in the flags will conform to this for-
      mat 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

	0 1	2 3	4 5	6 7	8 9 10 11	12 13 14 15	
0		DRQREQ	SBUF	IOWAKE	BLOCKED DONE DO'POST SERIAL'POST	CDT'QUEUED MOVE'DONE CUR'REQ DISABLE LDR'REQ LDR'INLOC	LDR'FLAGS
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.
(0:2)
            Not used.
(2:1) - DRQREQ
            Set if LDR causes a physical I/O.
(3:1) - SBUF
            Set if request is to/from a System Buffer.
(4:1) - IOWAKE
            Set if system should wake up the process
            when the logical I/O completes.
(5:1) - BLOCKED
            Set if the process wants to wait for the
            logical disc request to complete.
(6:1) - DONE
            Set when the logical disc request is complete
            and the process will be awakened (if IOWAKE is set)
(7:1) - DO'POST
            Set if the caller wants to be waited until the physical
            post to disc completes. Only valid for write requests.
(8:1) - SERIAL'POST
            Set when the physical post should be
            through the Global Serial Write queue.
(9:1) - CDT'QUEUED
            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) - MOVE'DONE
            The move has been completed, but
            the process won't be awakened until the DONE bit is set.
(11:1) - Not used.
(12:1) - CUR'REQ
            Set if this request is the current/active
            request.
(13:1) - DISABLE
            Set if the request is disabled.
(14:1) - LDR'REQ
            Set if this is a logical disc request.
(15:1) - LDR'INLOC
            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' 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 - Disc Free Space
        5-7 - 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 data 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.)

CHAPTER 24 NATIVE LANGUAGE SUPPORT

NL/3000 Internal Table Structure

```
NLS FILE CODES
LANGDEF.PUB.SYS   -  1228
CHRDEFxx.PUB.SYS  -  1229
NLSDEF.PUB.SYS    -  1229
```

Native Language Support (NLS) Table Overview
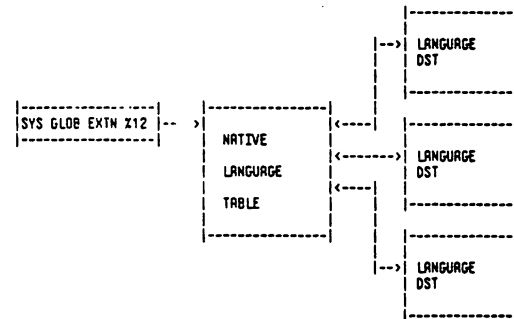
Native Language Table (NLT)

This table is created by INITNLS (called by PROGEN). The DST number is contained in SYSGLOB extension X12. The Native Language Table (NLT) contains the description of all the character sets needed to support the installed languages, and additional information needed to support the configured languages (DST numbers of the languages associated DSTs, character sets, etc.).

Every installed language has an associated Language DST, as set up by INITNLS.

```
|---------------------------|
|  NLT OVERHEAD TABLE       |
|---------------------------|
|  NLT INSTALLED LANGUAGE   |
|  TABLE                    |
|---------------------------|
|  NLT INSTALLED CHARACTER  |
|  SET TABLE                |
|---------------------------|
|  NLT CHARACTER ATTRIBUTE  |
|  TABLE                    |
|---------------------------|
```

NLT Overhead Table

The NLT overhead table is eight (8) words long.

```
 |-------------|-------------|
0|    "N"      |    "L"      |
1|    "T"      |    " "      |
 |-------------|-------------|
2| LENGTH OF NLT (IN WORDS)  |
 |---------------------------|
3| NUMBER INSTALLED LANGUAGES|
 |---------------------------|
4| NUMBER INSTALLED CHAR SETS|
 |---------------------------|
5| SYSTEM LANGUAGE ID NUMBER |
 |---------------------------|
6| SYSTEM LANGUAGE LDST      |
 | NUMBER                    |
 |---------------------------|
7| RESERVED                  |
 |---------------------------|
```

NLT Installed Language Table Format

For each supported non-NATIVE/3000 languages there is a 12-word language entry.

NLT Installed Character Set Table Format

For each character set installed on the system there is an 11 word character set table. It has the following format:

```
         |------------------------|
       0 | CHARACTER SET ID NUMBER |
         |------------------------|
       1 | CHARACTER SET TYPE      |
         |------------------------|
       2 | POINTER TO CHARACTER   |
         | ATTRIBUTES TABLE       |
         |------------------------|
       3 |                        |
         ~ CHARACTER SET NAME     ~
         | (8 WORDS)              |
     X12 |                        |
         |------------------------|
```

NLT Character Attributes Table

The NLT Character Attributes Table is comprised of a table for each configured character set. At this time, only two character sets are configurable: Class Four Languages (KANJI-based) and Nonclass Four Languages.

```
                                     CHARACTER TYPE = 1
 |------------------|          |------------------|
 | CHAR SET # 1     |--------->| 256 byte         |
 | ATTRIBUTES       |          | ATTRIBUTE        |
 |------------------|          | TABLE            |
 | CHAR SET # 2     |----|     |------------------|
 | ATTRIBUTES       |    |
 |                  |    |       CHARACTER TYPE = 2
 ~                  ~    |     |------------------|
 |                  |    |---->| 8 WORD KANJI     |
 |                  |          | ATTRIBUTE        |
 |------------------|          | TABLE            |
                               |------------------|
```
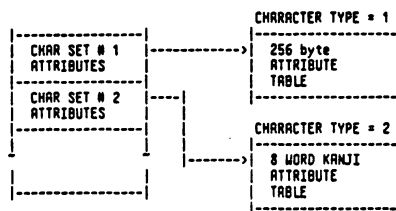
The type = 1 attribute table is a 256 byte table. Each byte corresponds to a character with that octal value.

```
Attribute  0    - Numeric character
           1    - Special character (i.e., "!", "?", ".")
           2    - Alphabetic uppercase character
           3    - Alphabetic lowercase character
           4    - Control code
           5    - Invalid character (unused code)
```

Language DST

For each language installed on a target system (with the exception of NATIVE-3000) INITNLS will build one language DST with the following structure:

```
 |----------------------------|
 | LDST OVERHEAD TABLE        |
 |----------------------------|
 | LDST TRANSLATION TABLES    |
 | (5 SUBTABLES)              |
 |----------------------------|
 | LDST CUSTOM DATA TABLES    |
 |----------------------------|
 : LDST NATIONAL SPECIAL      :
 : TABLE                      :
 : (AN OPTIONAL TABLE)        :
 |----------------------------|
```

LDST Overhead Table

The overhead region has the following format:

```
         |-------------|-------------|
       0 |    "L"      |    "D"      |
         |-------------|-------------|
       1 |    "S"      |    "T"      |
         |-------------|-------------|
       2 |    LDST SIZE IN WORDS     |
         |--------------------------|
       3 |    NLT DST NUMBER        |
         |--------------------------|
       4 |    LDST OFFSET TO CUSTOM |
         |    DATA TABLES          |
         |--------------------------|
       5 |    LDST OFFSET TO NATIONAL|
         |    SPECIAL TABLES       |
         |--------------------------|
       6 |                          |
         ~    RESERVED              ~
       7 |                          |
         |--------------------------|
```

The national special table is optional. If it does not exist, the pointer to it is zero.

LDST Translation Tables

For each language several translation tables are stored:

```
|-----------------------|
| LDST UPSHIFT TABLE     |
| (128 WORDS)           |
|-----------------------|
| LDST DOWNSHIFT TABLE   |
| (128 WORDS)           |
|-----------------------|
| LDST ASCII -> EBCDIC   |
| CONVERSION TABLE       |
| (128 WORDS)           |
|-----------------------|
| LDST EBCDIC -> ASCII   |
| CONVERSION TABLE       |
| (128 WORDS)           |
|-----------------------|
| LDST COLLATING         |
| SEQUENCE TABLE         |
| (CLASS DEPENDENT)      |
|-----------------------|
```

LDST Collating Sequence Table

The LDST Collating Sequence Table is of different formats depending upon the class of the language.

Overview

Class One
Languages: Some languages, namely American English and Katakana, can be collated by using the numerical representation of the ASCII encoding as the sequence number for any given character. These languages can use the Compare Bytes machine instruction.

Class Two
Languages: Some languages may be able to use the COBOLII machine instruction, Compare-Translated-Strings. These languages need to have a one-to-one mapping of character encoding to sequence number. Any algorithm for this class of language must take into account the fact that not every HP 3000 has COBOLII firmware.

Class Three
Languages: Many languages will not be able to use either of the tactics described above. There are a number of language-dependent algorithms that need to be supported.

Class Four
Languages: Some languages require 16-bit character string encoding. Collating these languages is not supported. The collating sequence table for this class of language is reserved.

Class One Languages

Since class one languages will use the compare bytes machine instruction (CMPB), the whole collating sequence table for this class is 3 words.

```
 |-----------------------|
0|          3            |
 |-----------------------|
1|    LANGUAGE ID        |
 |-----------------------|
2|    LANGUAGE CLASS     |
 |-----------------------|
```

Class Two Languages

This sequence table has a 13-word over head table and a 128-word sequence table.

```
  |-------------------------------------|
 0|               139                   |<----|
  |-------------------------------------|     |
 1|          LANGUAGE ID                |     |
  |-------------------------------------|     |
 2|          LANGUAGE CLASS             |     |
  |-------------------------------------|     |
 3|               11                    |     |
  |-------------------------------------|     |
 4|               128                   |     |
  |-------------------------------------|     |
 5|                0                    |     >-Overhead table
  |-------------------------------------|     |
 6|                0                    |     |
  |-------------------------------------|     |
 7|                0                    |     |
  |-------------------------------------|     |
10|                0                    |     |
  |-------------------------------------|     |
11| LOWEST CHAR.      | HIGHEST CHAR.   |     |
  | SEQUENCE VALUE    | SEQUENCE VALUE  |     |
  |-------------------------------------|     |
12|          RESERVED                   |<----|
  |-------------------------------------|
13| SEQUENCE # 0      | SEQUENCE # 1    |<----|
  |-------------------------------------|    |
14| SEQUENCE # 2      | SEQUENCE # 3    |    |
  |-------------------|-----------------|    >-Sequence table
  |         :         |        :        |    |
  |-------------------|-----------------|    |
212| SEQUENCE # 254   | SEQUENCE # 255  |<---|
  |-------------------------------------|
```

Note: Word X11 of the overhead contains in the left byte the character value, which has the lowest sequence number and in the right byte the character value, which has the highest sequence number.

In the 128-word sequence table, the byte value of the character is used as a byte pointer in the collating table.

The byte value of the character is used as a byte pointer collating entries.

Class Three Languages

```
  |-------------------------------------|
 0| TABLE LENGTH (WORDS)     |<----|
  |-------------------------------------|    |
 1| LANGUAGE ID              |         |
  |-------------------------------------|    |
 2| LANGUAGE CLASS           |         |
  |-------------------------------------|    |
 3|          11              |         |
  |-------------------------------------|    |
 4|         256              |         |
  |-------------------------------------|    |
 5| POINTER TO 2:1 MAPPING TABLE |     >-Overhead Table
  |-------------------------------------|    |
 6| LENGTH OF 2:1 MAPPING TABLE |      |
  |-------------------------------------|    |
 7| POINTER TO 1:2 MAPPING TABLE |     |
  |-------------------------------------|    |
10| LENGTH OF 1:2 MAPPING TABLE |      |
  |-------------------------------------|    |
11| LOWEST CHAR.    | HIGHEST CHAR.|    |
  | SEQUENCE VALUE  | SEQUENCE VALUE|   |
  |-------------------------------------|    |
12| RESERVED                 |<----|
  |-------------------------------------|
13<--| SEQUENCE ENTRY # 0   |<----|
  |-------------------------------------|    |
  | SEQUENCE ENTRY # 1       |         |
  |-------------------------------------|    >-Sequence table
  |          :               |         |
  |-------------------------------------|    |
370| SEQUENCE ENTRY # 255    |<----|
  |-------------------------------------|
  | 2:1 CHARACTER MAPPING TABLE |
  |-------------------------------------|
  | 1:2 CHARACTER MAPPING TABLE |
  |-------------------------------------|
  |----------------->|
```

Class Three Languages (Cont.)

```
                        |-----MODE------|
                               1 1 1 1 1 1
              0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
              |------------------|---|-----------|
              |        0         | 0 |     0     |  These characters will
              |------------------|---|-----------|  be ignored when sorted.
                            OR
              0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
              |------------------|---|-----------|
              |    SEQUENCE #     | 0 |     0     |  All 1:1 mapping
              |------------------|---|-----------|  characters
                                                    without case or
                                                    accent priority
                            OR
              0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
              |------------------|---|-----------|
              |    SEQUENCE #     | 0 |  PRIORITY |  1:1 mapping
              |------------------|---|-----------|  characters with
                                                    case or accent
                                                    priority
                            OR
              0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
              |------------------|---|-----------|
              |    SEQUENCE #     | 1 |   INDEX   |  2:1 mapping
              |------------------|---|-----------|  characters
                            OR
              0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
              |------------------|---|-----------|
              | SEQ. # OF 1. | 2 |   INDEX   |  1:2 mapping
              |------------------|---|-----------|  characters
```

The byte value of the character is used as an index to the sequence entries.

2:1 Character Mapping Table

```
                                                  1 1 1 1 1 1
                                0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
   |-------------------|        |------------------|-----------|
   | 2:1 MAPPING ENTRY |----> 0 |        0         | LEGAL CHAR 1 |
   |-------------------|        |------------------|-----------|
                             1  | SEQ.# OF PAIR    | MODE OF PAIR |
                                |------------------|-----------|
                                |                              |
        (N-1)*2                 |------------------|-----------|
                                |        0         | LEGAL CHAR N |
                                |------------------|-----------|
        (N-1)*2+1               | SEQ.# OF PAIR    | MODE OF PAIR |
           n*2                  |------------------|-----------|
                                |              -1              |
           N*2+1                |------------------|-----------|
                                |        MODE NON MATCH        |
                                |------------------------------|
```

```
                              |--------- MODE ----------|
          0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
         |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
         |--------------------------|-----|--------------|
                                       |        |
                                       |        |------ INDEX/TYPE
                                       |               PRIORITY
                                       |               (BITS 10:6)
                                       |
                                       |-------- MODE TYPE (0,1,OR
                                                 (BITS 8:2)
```
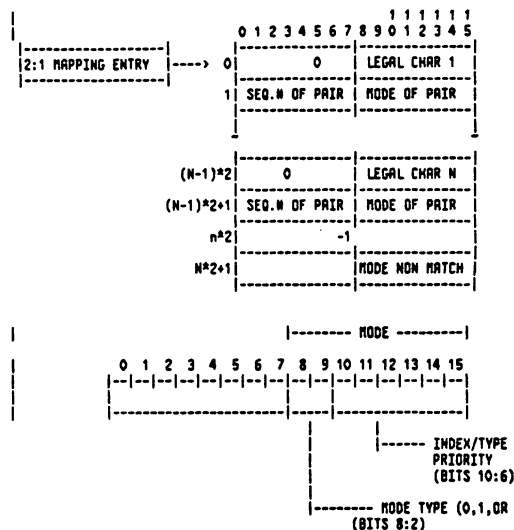
Entry has same format as mode options in the LDST Collating Sequence Table Format for Class Three Languages.

1:2 Character Mapping Table

```
                                                  1 1 1 1 1 1
                                0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
   |-------------------|        |------------------|-----------|
   | 1:2 MAPPING ENTRY |----> 0 | SEQ.# OF 2ND     | MODE OF BOTH |
   |-------------------|        |------------------|-----------|
                             1  | SEQ.# OF 2ND     | MODE OF BOTH |
                                |------------------|-----------|
                                |                              |
                                |------------------|-----------|
                             N  | SEQ.# OF 2ND     | MODE OF BOTH |
                                |------------------|-----------|
```

```
                              |-------- MODE ---------|
          0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
         |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
         |--------------------------|-----|--------------|
                                       |        |
                                       |        |------ INDEX/TYPE
                                       |               PRIORITY
                                       |               (BITS 10:6)
                                       |
                                       |-------- MODE TYPE (0,1,OR
                                                 (BITS 8:2)
```
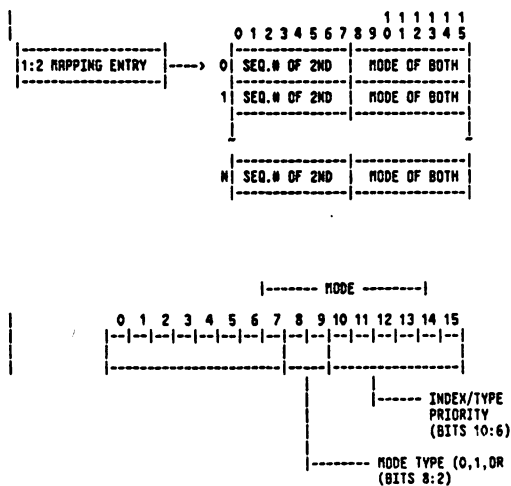
Entry has same format as one above.

Class Four Languages

Class four languages require 16-bit character encoding. Sorting in class four languages is not implemented in this release of NLS. A preliminary collating sequence table is planned to be 8 words in length.

```
         |-------------------------------|
       0 | ADDRESS ON DISC - PATTERN 1   |
         |-------------------------------|
       1 | ADDRESS ON DISC - PATTERN 2   |
         |-------------------------------|
         |                               |
         |-------------------------------|
       7 | ADDRESS ON DISC - PATTERN 8   |
         |-------------------------------|
```
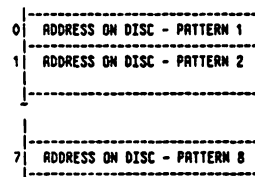
LDST Custom Data Table Format

This table is 196 words long. The formats and information in this table are language dependent, and may be modified with LANGINST.PUB.SYS.

```
    |-------------------------------|
  0 | LDST CALENDAR SKELETON        |0
    | (9 WORDS)                     |
    |-------------------------------|
 11 | LDST CUSTOM DATE SKELETON     |9
    | (13 BYTES)                    |
    |-------------------------------|
 20 | LDST TIME SKELETON            |16
    | (4 WORDS)                     |
    |-------------------------------|
 24 | LDST ABBREVIATED MONTH NAMES  |20
    | (24 WORDS)                    |
    |-------------------------------|
 54 | LDST FULL MONTH NAMES         |44
    | (122 WORDS)                   |
    |-------------------------------|
164 | LDST ABBREVIATED WEEKDAY NAMES|116
    | (21 BYTES)                    |
    |-------------------------------|
177 | LDST FULL WEEKDAY NAMES       |127
    | (42 WORDS)                    |
    |-------------------------------|
251 | LDST YES/NO CHARACTER STRINGS |169
    | (6 WORDS)                     |
    |-------------------------------|
257 | LDST THOUSANDS INDICATORS     |175
    | (1 WORD)                      |
    |-------------------------------|
260 | LDST CURRENCY SYMBOL          |176
    | (5 BYTES)                     |
    |-------------------------------|
263 | LDST RESERVED                 |179
    |-------------------------------|
```

LDST National Special Table

This table is optional and its existence is signaled by a nonzero pointer in the LDST overhead region. It is used to store data unique to a given language (e.g. the Emperor data for the Japanese calendar).

```
    |-----------------------|
    | LENGTH                |
    |-----------------------|
    | NATIONAL              |
    | DEPENDENT             |
    | DATA                  |
    |-----------------------|
```

Date Formats for Japan and Taiwan

For a given language, there is only one date format possible. The format of the year stored in the date format of the LDST can either be yyyy or yy for the Julian dates or Nyy for either the Japanese date (Emperor Era) or the Taiwanese date foundation of republic date).

If the format of the year stored as the date format in the LDST is Nyy then either the Japanese emperor dates or the Taiwanese foundation date has to be stored in the national dependent table.

National Dependent Table Formats

```
      |-------------------------|
   X0 | LENGTH OF TABLE(WORDS)  |
      |-------------------------|
   X1 |          ID             |
      |-------------------------|
   X2 | NUMBER OF ENTRIES       |
      |-------------------------|
   X3 | NUM OF HP SUPPLIED ENTR. |
      |-------------------------|
X4 + X5| PERIOD ENTRY 1         |
      |-------------------------|
      |        .                |
      |-------------------------|
(2n+2) +| PERIOD ENTRY n        |
(2n+3)|-------------------------|
```

The period entries are two word entries of the following format:

```
    0       6 7         15
    |----------|------------|
    | YEAR OF  | DAY OF THE |  WORD 1 (STARTING DATE)
    | CENTURY  | YEAR       |
    |----------|------------|

    0         7 8         15
    |-----------|-----------|
    | STARTING  | EMPEROR   |  WORD 2
    | YEAR      | SYMBOL    |       .
    |-----------|-----------|
```

The ID for Japanese and Taiwanese date formats is always set to 1.

Japanese Date Format

There are three entries which do not change. The user can add new entries. These entries have to be stored in ascending order sorted by word 1.

The values of the entries are:

	Starting Date (MDY)	Octal Value	Starting year	Emperor Symbol
*	1/ 1/1873	X1	X41	M
	7/30/1912	X14324	X1	T
	12/25/1926	X32547	X1	S

* Since this starting time is in the 19 th century and we are not able to handle dates before 1900 easily, we store X1 as starting time.

For new date entries created by the customer the starting year will always be 1.

Taiwanese Date Format

There are two entries for the Taiwanese national dependent table.

The values of the entries are:

Starting Date (MDY)	Octal Value	Starting Year	Emperor Symbol
1/ 1/1900	X1	X0	X40
1/ 1/1912	X14001	X1	X40

The user does not need to add new entries.

CHAPTER 25 ATP/ATP37/ADCC

Overview

This chapter contains a description of the monitor/printer DIT and tables in the terminal data segment (TDS) used by the MPE V/E ATP/ATP37 and ADCC terminal drivers.

Terminal Data Segment Formats

ATP/ATP37 Terminal Data Segment Format

```
        0  1  2  3  4  5  6  7  8  9  10 11 12 13 14 15
       |---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
     0|
    42| TERMINAL DATA SEGMENT HEADER - %43 WORDS                       |
      |----------------------------------------------------------------|
    43|
   242| HARDWARE DIT POINTER TABLE - %200 WORDS                        |
      |----------------------------------------------------------------|
   243|
   646| VFC TABLE - %404 WORDS                                         |
      |----------------------------------------------------------------|
   647|
   732| ATP HARDWARE DIT FOR UNIT 0 - %64 WORDS                        |
      |----------------------------------------------------------------|
   733|
   752| CONTROL PROGRAM AREA FOR UNIT 0 - %20 WORDS                    |
      |----------------------------------------------------------------|
   753|
   767| PROTOCOL AND DATA MANAGER FIXED DIT FOR UNIT 0 - %15 WORDS     |
      |----------------------------------------------------------------|
   770|
  1103| PORT PROTOCOL DIT FOR UNIT 0 - %114 WORDS                      |
      |----------------------------------------------------------------|
  1104|
  1136| PROTOCOL AND DATA MANAGER VARIABLE DIT FOR UNIT 0 - %33 WORDS  |
      |----------------------------------------------------------------|
```

ATP/ATP37 Terminal Data Segment Format (Cont.)

```
      |----------------------------------------------------------------|
  1137|
  1426| DIT'S FOR UNIT 1 - %270 WORDS                                  |
      |----------------------------------------------------------------|
      |
      | DIT'S FOR UNIT N - %270 WORDS                                  |
      |----------------------------------------------------------------|
      |
      | PCC DUMP AREA - %400 WORDS                                     |
      |----------------------------------------------------------------|
      |
      | MESSAGE TABLE - %53 WORDS                                      |
      |----------------------------------------------------------------|
      |
      | PORT ERROR DUMP AREA - %2424 WORDS                             |
      |----------------------------------------------------------------|
      |
      | TBUF TABLE HEADER - %12 WORDS                                  |
      |----------------------------------------------------------------|
      |
      | TBUF'S - %105 WORDS EACH                                       |
      |----------------------------------------------------------------|
```

Fixed overhead is %3760 words. Overhead per logical device is %270 words. Each TBUF is %105 words.

For ATP37s there is one data segment created for each channel (DRT) that contains an ATP37. For ATPs there is at least one, maybe two data segments created for each channel (DRT) that contains an ATP. Devices configured between units 0/47 are in one data segment and if devices are configured between units 48/95 they are in a second data segment. The addresses for the data segment(s) are contained in the first six words of the ILTX for the channel. If there is a second ATP data segment then the data segment address is in the second six words of the ILTX.

ADCC Terminal Data Segment Format

```
        0  1  2  3  4  5  6  7  8  9  10 11 12 13 14 15
       |---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
     0|
    42| TERMINAL DATA SEGMENT HEADER - %43 WORDS                       |
      |----------------------------------------------------------------|
    43|
   242| HARDWARE DIT POINTER TABLE - %200 WORDS                        |
      |----------------------------------------------------------------|
   243|
   646| VFC TABLE - %404 WORDS                                         |
      |----------------------------------------------------------------|
   647|
  1010| ADCC HARDWARE DIT FOR UNIT 0 - %142 WORDS                      |
      |----------------------------------------------------------------|
  1011|
  1025| PROTOCOL AND DATA MANAGER FIXED DIT FOR UNIT 0 - %15 WORDS     |
      |----------------------------------------------------------------|
  1026|
  1141| PORT PROTOCOL DIT FOR UNIT 0 - %114 WORDS                      |
      |----------------------------------------------------------------|
  1142|
  1174| PROTOCOL AND DATA MANAGER VARIABLE DIT FOR UNIT 0 - %33 WORDS  |
      |----------------------------------------------------------------|
  1175|
  1522| DIT'S FOR UNIT 1 - %326 WORDS                                  |
      |----------------------------------------------------------------|
      |
      | DIT'S FOR UNIT N - %326 WORDS                                  |
      |----------------------------------------------------------------|
```

ADCC Terminal Data Segment Format (Cont.)

```
      |----------------------------------------------------------------|
      |
      | MESSAGE TABLE - %53 WORDS                                      |
      |----------------------------------------------------------------|
      |
      | PORT ERROR DUMP AREA - %2424 WORDS                             |
      |----------------------------------------------------------------|
      |
      | TBUF TABLE HEADER - %12 WORDS                                  |
      |----------------------------------------------------------------|
      |
      | TBUFS - %105 WORDS EACH                                        |
      |----------------------------------------------------------------|
```

Fixed overhead is %3360 words. Overhead per logical device is %326 words. Each TBUF is %105 words.

For ADCCs there is one data segment created for all configured ADCC devices. It will contain the tables for a maximum of 64 configured devices. The address of the data segment is in the first six words of the ILTX.

Terminal Data Segment Tables

Terminal Data Segment Header Format

```
         0  1  2  3  4  5  6  7  8  9  10 11 12 13 14 15
        |---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
    X0| TERMINAL DATA SEGMENT VERSION NUMBER                              |
        |------------------------------------------------------------------|
     1| TDS RELATIVE POINTER TO HARDWARE DIT TABLE                        |
        |------------------------------------------------------------------|
     2| TDS RELATIVE POINTER TO TBUF TABLE                                |
        |------------------------------------------------------------------|
     3| TDS RELATIVE POINTER TO VFC TABLE                                 |
        |------------------------------------------------------------------|
     4| TDS RELATIVE POINTER TO 1ST HARDWARE DIT                          |
        |------------------------------------------------------------------|
     5| TDS RELATIVE POINTER TO 1ST DEVICE WAITING FOR A TBUF            |
        |------------------------------------------------------------------|
     6| TDS RELATIVE POINTER TO LAST DEVICE WAITING FOR A TBUF           |
        |------------------------------------------------------------------|
     7| DRT NUMBER OF CHANNEL - ATP/ATP37 ONLY                            |
        |------------------------------------------------------------------|
    10| LOWEST LOGICAL DEVICE NUMBER IN TDS                               |
        |------------------------------------------------------------------|
    11| HIGHEST LOGICAL DEVICE NUMBER IN TDS                              |
        |------------------------------------------------------------------|
    12| AIB POLL MASK - ATP/ATP37 ONLY                                    |
        |------------------------------------------------------------------|
    13| TDS RELATIVE POINTER TO MESSAGE TABLE                             |
        |------------------------------------------------------------------|
    14| TDS RELATIVE POINTER TO PCC DUMP AREA - ATP/ATP37 ONLY            |
        |------------------------------------------------------------------|
    15| TDS RELATIVE POINTER TO PORT ERROR DUMP AREA                      |
        |------------------------------------------------------------------|
    16| LOWEST LOGICAL DEVICE CONFIGURED ON CHANNEL - ATP/ATP37 ONLY      |
        |------------------------------------------------------------------|
    17| HIGHEST LOGICAL DEVICE CONFIGURED ON CHANNEL - ATP/ATP37 ONLY     |
        |------------------------------------------------------------------|
    20| STATUS WORD OF TDS WHEN IT WAS BUILT - ATP ONLY                   |
        |------------------------------------------------------------------|
    21| STATUS WORD OF TDS WHEN IT WAS INITIALIZED - ATP ONLY             |
        |------------------------------------------------------------------|
    22| STATUS WORD FOR DIAGNOSTICS                                       |
        |------------------------------------------------------------------|
    23|                                                                    |
    24| VERSION NUMBER OF LPMON                                           |
        |------------------------------------------------------------------|
```

Terminal Data Segment Header Format (Cont.)

```
        |------------------------------------------------------------------|
    25|                                                                    |
    26| VERSION NUMBER OF TERMMON                                         |
        |------------------------------------------------------------------|
    27|                                                                    |
    30| VERSION NUMBER OF PHYSICAL DRIVER - ATPDRIVER/ADCCDRIVER          |
        |------------------------------------------------------------------|
    31|                                                                    |
    32| VERSION NUMBER OF IHANDLER                                        |
        |------------------------------------------------------------------|
    33|                                                                    |
    34| VERSION NUMBER OF INITIALIZATION PROCEDURE-ATP/ATP37/ADCCINIT     |
        |------------------------------------------------------------------|
    35|                                                                    |
    36| VERSION NUMBER OF IMANAGER                                        |
        |------------------------------------------------------------------|
    37|                                                                    |
    40| VERSION NUMBER OF TERMUTIL                                        |
        |------------------------------------------------------------------|
    41|                                                                    |
    42| VUUFF NUMBER OF SOFTWARE                                          |
        |------------------------------------------------------------------|
```

Hardware DIT Pointer Table Format

ATP/ATP37 Hardware DIT Pointer Table Format

```
         0  1  2  3  4  5  6  7  8  9  10 11 12 13 14 15
        |---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
     0| TDS RELATIVE POINTER TO ATP/ATP37 UNIT 0 HARDWARE DIT            |
        |------------------------------------------------------------------|
     1| TDS RELATIVE POINTER TO ATP/ATP37 UNIT 1 HARDWARE DIT            |
        |------------------------------------------------------------------|
      |                                                                    |
      |                                                                    |
   177| TDS RELATIVE POINTER TO ATP/ATP37 UNIT 127 HARDWARE DIT          |
        |------------------------------------------------------------------|
```

This table, words X43/X242 in the data segment, contains a data segment relative pointer to the hardware DIT for each unit configured. If the unit is not configured the pointer will be a minus one.

ADCC Hardware DIT Pointer Table Format

```
         0  1  2  3  4  5  6  7  8  9  10 11 12 13 14 15
        |---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
     0| DRT NUMBER OF 1ST DEVICE                                          |
        |------------------------------------------------------------------|
     1| TDS RELATIVE POINTER TO 1ST ADCC HARDWARE DIT                    |
        |------------------------------------------------------------------|
     2| DRT NUMBER OF 2ND DEVICE                                          |
        |------------------------------------------------------------------|
     3| TDS RELATIVE POINTER TO 2ND ADCC HARDWARE DIT                    |
        |------------------------------------------------------------------|
      |                                                                    |
      |                                                                    |
   176| DRT NUMBER OF LAST DEVICE                                         |
        |------------------------------------------------------------------|
   177| TDS RELATIVE POINTER TO LAST ADCC HARDWARE DIT                   |
        |------------------------------------------------------------------|
```

This table, words X43/X242 in the data segment, contains a data segment relative pointer to the hardware DIT for each device configured. The table also contains the DRT number of the device. If the device is not configured the pointer and DRT number will be a minus one.

VFC Table Format

```
         0  1  2  3  4  5  6  7  8  9  10 11 12 13 14 15
        |---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
    -4|                                                                    |
    -1| VFC TABLE SIR                                                     |
        |------------------------------------------------------------------|
     0|                                                                    |
    17| VFC ENTRY 0                                                       |
      |                                                                    |
      |                                                                    |
        |------------------------------------------------------------------|
   360|                                                                    |
   377| VFC ENTRY 15                                                      |
        |------------------------------------------------------------------|
```

VFC Entry Format

```
         0  1  2  3  4  5  6  7  8  9  10 11 12 13 14 15
        |---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
     0|                                                                    |
     3| VFC FILE NAME                                                     |
     4|                                                                    |
        |------------------------------------------------------------------|
     7| VFC FILE GROUP NAME                                               |
    10|                                                                    |
        |------------------------------------------------------------------|
    13| VFC FILE ACCOUNT NAME                                             |
        |------------------------------------------------------------------|
    14| REFERENCE COUNT                                                   |
        |------------------------------------------------------------------|
    15| TDS RELATIVE POINTER TO TBUF CONTAINING INITIALIZATION STRING    |
        |------------------------------------------------------------------|
    16| TDS RELATIVE POINTER TO TBUF WITH VFC 0/VFC 7                    |
        |------------------------------------------------------------------|
    17| TDS RELATIVE POINTER TO TBUF WITH VFC 8/VFC 15                   |
        |------------------------------------------------------------------|
```

Protocol and Data Manager DITs Format

Protocol and Data Manager Fixed DIT Format

```
     0  1  2  3  4  5  6  7  8  9  10 11 12 13 14 15
    |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
  0| TERMINAL DST RELATIVE POINTER TO HARDWARE DIT   |
    |--------------|--------|---|---|-------|--|-----|
  1|              |CONTROL |SPD|LIN|CONNECT|  |BRO|PWF|
   | PORTSTATE    |TYPE    |SPC|TYP|TYPE   |  |KEN|FLG|
    |--------------|--------|---|---|-------|--|-----|
  2| CARRIER FAIL CNT    |  | PS| MODEM SIGNALS STATE |
    |---------------------------------------|---------|
  3| CARRIER FAIL TIMER                               |
    |-------------------------------------------------|
  4| 2631B XON TIMER                                  |
    |-------------------------------------------------|
  5| DATA SET READY TIMER                             |
    |-------------------------------------------------|
  6| DATA TERMINAL READY TIMER                        |
    |-------------------------------------------------|
  7| PORT SPEED IN CHARACTER PER SECOND               |
    |-------------------------------------------------|
 10| TERMINAL DST RELATIVE PORT PROTOCOL POINTER    |8
    |---|---|-------|------|------|---|-----|---------|
 11|CHR|PTY| WRITE | READ | READ |CLR|     | MODEM  |9
   |SIZ|ENB| PARITY| PARITY| ALT CHARS |FLD|  | STATE |9
    |---|---|-------|------|------|---|-----|---------|
 12| LINE SPEED IN CHRS/SEC - (DEFAULT)             |10
    |------------------------|------------------------|
 13| DEFAULT TERMINAL TYPE NUMBER | CURRENT TERMINAL TYPE NUMBER |11
    |------------------------|------------------------|
 14| HARDWARE TYPES                                 |12
    |-------------------------------------------------|
```

WORD 0

PD'IODITP

Terminal DST relative pointer to the hardware DIT.

WORD 1

PD'PORTSTATE(1).(0:4)

Indicates the current state of the hardware. This field is set after "started" is returned from the physical driver.

0 - Unused.
1 - Reading. The hardware is transferring data from the device to main memory.
2 - Writing. The hardware is transferring data from main memory to the device.
3 - Idle read. The driver is doing a "dummy read" while waiting for the next operation. Only special characters will be processed.
4 - Input save. The hardware is currently idle. The hardware is saving read characters to be processed against the next read, write or idle read. The Interrupt Manager cannot be called by the physical driver when in the input save state.
5 - Unused.
6 - Unused.
7 - Unused.
8 - Unused.
9 - Selftest. The hardware is currently doing a selftest.
10- Speedsensing. The hardware is currently waiting to speedsense the port.
11- Set port protocol. The port protocol is currently being set up. This includes:

- XON/XOFF enable/disable
- 7-bit/8-bit characters
- FF enable/disable
- Input parity enable/disable
- Output parity type
- ENQ/ACK characters and block count
- Delay characters and delay count
- Linespeed

12- Set special characters. The read, read secondary, write, and write edit special character sets are currently being set up.
13- Modem control. The modem interface, input, and output control lines are currently being set up.
14- Unused.
15- Unused.

PD'CONTROLLER(1).(4:2)

Indicates the type of controller.

0 - Unused.
1 - ATP.
2 - ADCC.
3 - Unused.

PD'SPEED'SPECIFIED(1).(6:1)

Indicates if the device is to be a speed-specified port.

0 - Speedsensed.
1 - Speed-specified.

PD'LINETYPE(1).(8:1)

Indicates the type of line. Currently unused.

0 - Asynchronous.
1 - Synchronous.

PD'CONNECTTYPE(1).(10:2)

Indicates the type of connection.

0 - Direct connect. Device configured as subtype 0 or 14.
1 - Modem connect. Device configured as subtype 1 or 15.
2 - Modem connect. Device configured as subtype 9 or 13. This subtype is for CCITT type modems.

PD'BROKEN(1).(14:1)

Indicates if the port is broken.

0 - Port is not broken.
1 - Port is broken.

PD'POWERFAIL(1).(15:1)

Indicates if a power fail has just occurred.

0 - No power fail.
1 - Power fail has just occurred.

WORD 2

PD'CF'CNT(2).(0:6)

Indicates the number of times DCD has gone off during a read. This is incremented by the Interrupt Manager each time there is a carrier fail during the read. If carrier fails more that 50 times during the read the modem will be disconnected.

PD'PENDING'START(2).(7:1)

Indicates if there is a pending request.

0 - No pending request.
1 - Indicates that a pending operation needs to be started by the physical driver. This occurs only on modem ports when the physical driver was in the middle of processing a request and there was a carrier fail. The current request can not be started until carrier detect is back on. Pending is then returned to the P&D Manager. Once carrier goes back on the Interrupt Manager will start the pending operation.

PD'MODEM'SIGNALS(2).(8:8)

Indicates the last known state of the modem input signals. A zero (0) indicates the signal is on and a one (1) indicates the signal is unused or off.

Bit 0 - Unused.
Bit 1 - Clear to send.
Bit 2 - Signal quality.
Bit 3 - Data set ready.
Bit 4 - Call origin status.
Bit 5 - Secondary carrier detect.
Bit 6 - Ring indicator.
Bit 7 - Carrier detect.

WORD 3

PD'CFTIMER

Contains a carrier fail timer index. This is a 30-second timer that is started by the Interrupt Handler when a carrier fail occurs. If carrier detect does not come back on within 30 seconds the modem is disconnected.

WORD 4

PD'XONTIMER

Contains an XON timer index. For term types 21 and 22 (HP 2631B processing) a 60-second timer is started by the Interrupt Handler when the XOFF is received. If the XON is not received within the 60 seconds the "LDEV NOT READY" message is printed by the Initiation Manager.

WORD 5

PD'DSRTIMER

Contains a data set ready timer index. If the port is to be speedsensed a 2-minute timer is started by the Interrupt Handler when DSR goes on. A speedsense must then be completed within the 2-minutes. If the port is being FOPEN'ed then the timer is started by the Initiation Manager. DSR and DCD must come on and the modem connected within the 2-minutes.

WORD 6

PD'DTRTIMER

Contains a data terminal ready timer index. This is a 5-second timer started by the Initiation Manager when a modem is to be disconnected. DTR is driven low for 5 seconds to disconnect the modem.

WORD 7

PD'PORTSPEED

Indicates the current line speed in characters-per-second.

WORD 8

PD'PPROTOCOL

Terminal DST relative pointer to the port protocol table.

WORD 9

PD'CHARSIZE(9).(0:1)

Indicates the size of the data character.

    0 - 7-bit characters with a parity bit.
    1 - 8-bit characters.

PD'PARITYENAB(9).(1:1)

Indicates the state of parity checking.

    0 - Parity checking disabled.
    1 - Parity checking enabled.

PD'WPARITY(9).(2:2)

Indicates the type of parity generation for writes.

    0 - Space, high order bit forced to zero.
    1 - Mark, high order bit forced to one.
    2 - Even parity.
    3 - Odd parity.

PD'RPARITY(9).(4:2)

Indicates the type of parity for read characters. Types are the same as PD'WPARITY.

PD'ALL'PARITY(9).(1:5)

A reference to the above 5-bit parity field.

PD'ALTCHARSET(9).(6:3)

Indicates the contents of the read alternate character set.

    0/2 - Unused.
       3 - Idle read set.
       4 - Transparent read set.
       5 - View read set.
       6 - Binary read set.
       7 - Unused.

PD'CLEARF(9).(9:1)

Indicates if flow control waits should be aborted before the next write is started. Flow controls are aborted by the Initiation Manager after break or subsystem break has been accepted by the monitor.

        0 - Don't abort any flow controls.
        1 - Flow controls should be aborted.

PD'MODEM'STATE(9).(12:4)

Indicates the current state of a modem.

    0 - Unused.
    1 - Data set ready sensing. The Interrupt Manager is waiting
        for DSR to come on. After DSR is on the Interrupt Manager
        will wait for DCD to come on. The modem state will then be
        set to 3.

---

    2 - Data set ready sensing for FOPEN. This is the same as
        state 1 except that the port will be FOPEN'ed instead of
        speedsensed.
    3 - Data carrier detect sensing. The Interrupt Manager is
        waiting for DCD to come on. DCD must come on within
        30 seconds after DSR is on otherwise the modem will be
        disconnected. After DCD is on the port will be speed-
        sensed and the modem state will then be set to 5.
    4 - Data carrier detect sensing for FOPEN. This is the
        same as state 3 except the port is being FOPEN'ed.
    5 - Speedsensing. The modem has been connected and a speed-
        sense has been started.
    6 - Connected. The modem is connected and the port has been
        successfully speedsensed if FOPEN'ed. If DCD goes off
        once the modem is connected it must come back on within
        30 seconds or the modem is disconnected. If DSR goes
        off the modem is disconnected.
    7 - Disconnecting. The modem is being disconnected. The
        driver will drop DTR for 5 seconds. Then the modem
        state will go back to 1 or 2.

WORD 10

PD'DPORTSPEED

Indicates the configured line speed in characters-per-second.

WORD 11

PD'PPENTRYNUMB(11).(0:8)

Indicates the term type number as specified in the I/O configuration. If a term type file name was specified instead of a number or there was an error in trying to use the file name this field will default to 31.

PD'TERMTYPE(11).(8:8)

Indicates the current term type number. If a term type file name is being used this field will be a 0. The current term type number is the one returned for FCONTROLs.

WORD 12

PD'HARDWARE'TYPE

A controller dependent word. For ATP it contains the results of the selftest. The contents of the word are the same as HW'SELFTEST in the ATP physical driver DIT. For ADCC controllers the word is unused.

Protocol and Data Manager Variable DIT Format

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
0	SYS DB RELATIVE POINTER TO LOGICAL MONITOR DIT																
1	LDPSTATE				RED ECH	BRK RED	SS BRK	SBF RC	SBUF1 STATUS	DSC DEV	SBUF2 STATUS	DO SRQ	WWC	OC			
2	HEAD TBUF POINTER / SYS BUF 1 POINTER / BANK NUMBER																
3	TAIL TBUF POINTER / SYS BUF 2 POINTER / BANK OFFSET																
4	HEAD TBUFOFFSET / SYSBUF OFFSET / FROZEN DATA SEG OFFSET																
5	TAIL TBUFOFFSET / SYSBUF OFFSET																
6	BTANKED - NUMBER OF BYTES TANKED FOR WRITE																
7	READCNT - BYTE COUNT FOR READ																
10	IOQ EOR CHARACTER								NO LF	VEW RED	OWN RED	BINARY READ		EOFCODE			8 8
11	ALTERNATE EOR								ALTERNATE SSBREAK								9
12	LAST EOR																10
13	NEW TOP	NEW LIN	CON MOD	BRK MOD	CHR SET	FIL ING	SSB ENA	CON ENA	BRK ENA	CRI WRI	WAI TBF	PCC XON					11
14	START READ TIME / COMPUTER READ TIME 100THS																12
15	START READ TIME / READ TIME OUT VALUE																13
16	BRKTBUF - BROKEN READ HEAD TBUF POINTER																14
17	BRKRCNT - BROKEN READ COUNT																15
20	SPOOLED DISC ADDR / DEVICE LINK FOR TBUFS																16
21	SPOOLED DISC ADDR / SAVED "WAIT" TBUF POINTER																17
22	VEEPCNT - BYTES READ / TBUFS IN USE																18

Protocol and Data Manager Variable DIT Format (Cont.)

```
|----------------------------------------------------------------|
23| OPERATION ERROR - INTERRUPT MANAGER TO INITIATION MANAGER     |19
|---|---|---|---|---|---|---|---|---|------------------|---|---|---|
24|SUS|CTL|SPD|ESC|DC2|XON|LBL|BLK|        PEND      |31B|BIN|LOG|20
|LDP| X |SEM|PAR|RED|WAT|MOD|MOD|      LOPSTATE    |RST|MOD|DEV|
|---|---|---|---|---|---|---|---|------------------|---|---|---|
25| READ TYPE    | LLDMC    |PRT|RED LOC| LAST LDM OPCODE      |21
|--------------|----------|---|-------|---------------------|
26| REQUESTED DEVICE STATUS                                    |22
|----------------------------------------------------------------|
27| OLD TRANSFER COUNT / STATUS WRITE COUNT                    |23
|----------------------------------------------------------------|
30| POINTER TO SAVED EOF TBUF                                  |24
|----------------------------------------------------------------|
31| COUNT OF DATA IN SAVED EOF TBUF                            |25
|----------------------------------------------------------------|
32| READ TIMER INDEX                                           |26
|----------------------------------------------------------------|
```

WORD 0

PD'LDMDIT

SYS DB relative offset to the logical device monitor DIT.

WORD 1

NOTE:

This word is write shared between the Initiation Manager and Interrupt Manager. When the Initiation Manager is modifying this word the interrupt system should be off.

PD'LOPSTATE(1).(0:4)

Indicates the current logical operation state of a request.

0 - No operation. The driver has finished a request and is waiting for the next request.
1 - Reading.
2 - Writing.
3 - Status request to a device. The driver is currently writing a status request sequence to an HP 2631B printer.
4 - Status read. The driver is currently reading status back from an HP 2631B.
5 - Control-X response. The driver is currently writing the "!!!,cr,lf" after receiving the Control-X.
6 - Waiting for a carriage return. The driver is currently waiting for a carriage return to start or terminate a block mode read.
7 - Write with pending read. The driver is currently writing and at the completion of the write a read should be started.
8 - Hardware setup with pending read. The driver is currently setting up the hardware and when done it should start the pending read.
9 - Write with pending status request. The driver is currently writing and at the completion of the write should request status from the HP 2631B printer.
10- Speedsensing.
11- Set port protocol. The driver is currently setting up the current port protocol. This will include:
    - Enable/disable ENQ/ACK handshake
    - Enable/disable XON/XOFF handshake
    - ENQ/ACK handshake characters
    - ENQ/ACK block count
    - Delays for CR, LF, and FF
12- Set special characters. The driver currently setting up the read, the read alternate, the write, and the write edit special character sets.
13- Modem control. The driver is setting up the modem logic and is waiting for the modem signals to go to a known state.
14- Freezing. The driver is waiting for the users stack to be frozen before starting the current read. Not currently used.
15- View read set up. The driver is writing out the sequence to home the cursor and lock the key board before starting the current read.

PD'NO'READECHO(1).(4:1)

Indicates the current state of echo.

0 - Echo is enabled. Echo may be enabled by an IOQ function code of 8 (FCONTROL 12) by the Initiation Manager or an "ESC semicolon" from the device by the Interrupt Manager.
1 - Echo is disabled. Echo may be disabled with an IOQ function code of 9 (FCONTROL 13) by the Initiation Manager or by an "ESC colon" from the device by the Interrupt Manager.

PD'BREAK(1).(5:1)

Indicates if break has been detected.

0 - Break has not been detected.
1 - Break is enabled and has been read. This is set by the Interrupt Manager when break is detected and cleared by the Initiation Manager when break has been accepted or rejected.

PD'SSBREAK(1).(6:1)

Indicates if subsystem break has been detected.

0 - Subsystem break has not been detected.
1 - Subsystem break is enabled and has been detected. This is set by the Interrupt Manager when the subsystem break is read and cleared by the Initiation Manager when the subsystem break is accepted or rejected.

PD'SBUFREADCOMP(1).(7:1)

Indicates if software is currently processing a read byte count exhausted interrupt for a spooled read.

0 - Read complete interrupt is not currently being processed for a spooled read.
1 - Read complete is currently being processed for a spooled read.

PD'SBUF1'STAT(1).(8:2)

If the current read is using system buffers this field contains the status of one of the two system buffers used for the read.

0 - Empty. Available for the read.
1 - Filling. Currently being used for the read.
2 - Full. The buffer is full.

PD'DISCNCT'DEV(1).(10:1)

Indicates that the port is being disconnected so some error conditions will be ignored while this is taking place.

0 - No disconnect in progress.
1 - Disconnect in progress.

PD'SBUF2'STAT(1).(11:2)

If the current read is using system buffers this field contains the status of the second of the two system buffers used for the read.

0 - Empty. Available for the read.
1 - Filling. Currently being used for the read.
2 - Full. The buffer is full.

PD'DO'STATREQ(1).(13:1)

Indicates if an HP 2631B status request should be done after all the data tanked for the write has been written out.

0 - Don't request printer status.
1 - Request printer status at completion of the write.

PD'WWC(1).(14:1)

If set then the write is a non-critical write and logical device monitor is awakened at the completion of the write. The bit is set by the Initiation Manager and cleared by the Interrupt Manager.

PD'LOPCOMPLETE(1).(15:1)

Indicates if the current logical operation is complete.

0 - The logical operation is not complete.
1 - The logical operation is complete. Set by the Interrupt Manager at the end of the operation, i.e., read is complete. The logical monitor is then awakened and notified the operation is complete.

WORD 2

This word contains an address for the current data transfer. There are three types of data transfers with the current type indicated in PD'READ'LOC. The type of transfer indicates the type of address.

PD'HEADTBUF

A TDS relative pointer to the head TBUF.

PD'SBUF1

A SYS DB relative pointer to a system buffer. System buffers are used for spooled reads where the transfers are done in disc sector sizes (128 words). There are two system buffers used, the second in PD'SBUF2, for the read. When one becomes full the second one is used while the Initiation Manager transfers the data from the buffer to the disc. This swing buffer process is done until the read is complete.

PD'BANKNUMB

Contains the bank number of the data segment used for frozen reads.

WORD 3

This word contains an address for the current data transfer. There are three types of data transfers with the current type indicated in PD'READ'LOC. The type of transfer indicates the type of address.

PD'TAILTBUF

A TDS relative pointer to the tail TBUF.

PD'SBUF2

A SYS DB relative pointer to a system buffer. See PD'SBUF1 for information on how system buffers are used.

PD'BANKOFFSET

Contains the offset into a bank of the data segment used for frozen reads.

WORD 4

PD'HEADOFFSET

Contains a byte offset into the head TBUF or system buffer. For reads this is an offset to the first byte read. For writes this is an offset to the next byte to be written.

PD'ABSOFFSET

Contains an offset into the frozen data segment of where the spooled read data should be saved.

WORD 5

PD'TAILOFFSET

Contains a byte offset into the tail TBUF or system buffer. For reads this offset is to the last byte read. For writes this is an offset to the last byte to be written.

WORD 6

PD'BTANKED

This indicates the number of bytes that have been tanked into the TBUF for the write. If -1, then all bytes have been tanked. The Initiation Manager will tank up to a maximum of 5 TBUFs of write data before the write is started. As each TBUF is emptied the Interrupt Manager will restart the write. When there are 2 TBUFs left the Initiation Manager is notified and the tanking will resume, while the last two TBUFs are being emptied. This continues until all the data has been tanked by the Initiation Manager, and all the data written out by the Interrupt Manager.

WORD 7

PD'READCNT

This indicates the number of bytes for the current or pending read.

WORD 8

PD'IOQEOR(8).(0:8)

Contains the end-of-record character in the current read IOQ.

PD'NOLF(8).(8:1)

Indicates if a LF should be sent out at the end of the read.

    0 - Send a LF when the read is terminated by an EOR character.
    1 - Don't send a LF when the read is terminated by an EOR
        character.

PD'VIEWREAD(8).(9:1)

Indicates if the current read is a View read.

    0 - The current read is not a View read.
    1 - The current read is a View read. When a DC2 is
        received the Interrupt Manager will write out the
        sequence to home the cursor and lock the key board.

PD'OWNREAD(8).(10:1)

Indicates if special processing is done on DC2s received during the read.

    0 - DC2 characters are processed "normally", and will start
        block mode transfers.
    1 - DC2s don't start block mode transfers and will be saved
        as normal read data.

PD'BINARYREAD(8).(11:2)

Indicates if the read is a binary read.

    0 - The read is not binary.
    1 - The read is a binary read.

PD'EOF(8).(13:3)

End-of-file code. Not currently used.

PD'READFLAGS

A reference to all of the above fields.

WORD 9

PD'ALTEOR(9).(0:8)

Contains the EOR character as specified in FCONTROL 41.

PD'ALTSSBREAK(9).(8:8)

Contains the alternate subsystem break character. The alternate subsystem break character is not deleted from the read when detected.

PD'TRANSPARENT

A reference to both alternate characters. If non-zero then the read is known as a transparent read.

PD'ALTCHARS

A reference to both alternate characters.

WORD 10

PD'LASTEOR(10).(0:8)

This contains the last EOR character. When EOR characters are to be changed by the Initiation Manager this field will indicate if a physical change should be done, i.e., if the new EOR and old EOR are the same.

WORD 11

NOTE:

This is a write only word for the Initiation Manager. The Interrupt Manager should not write to any (used or unused) fields as it may cause software problems.

PD'NEWTOP(11).(0:1)

Indicates if the device is at top of form.

    0 - Device is not at top of form.
    1 - Device is at top of form.

PD'NEWLINE(11).(1:1)

Indicates if the device carriage is at the beginning of a new line. Note that this field is currently unused.

PD'CONSMODE(11).(2:1)

Indicates if the device is currently in console mode.

    0 - Device is not in console mode. Console mode can be
        cleared with an IOQ function code of 31 by the
        Initiation Manager.
    1 - Device is in console mode. Console mode can be set
        with an IOQ function code of 31 by the Initiation
        Manager.

PD'BREAKMODE(11).(3:1)

Indicates if the device is currently in breakmode.

    0 - Device not in breakmode. Cleared with an IOQ function
        code of 30 by the Initiation Manager.
    1 - Device is in breakmode. Set with an IOQ function code
        code of 30 by the Initiation Manager.

PD'CHARSET(11).(4:1)

Indicates what read special character set is currently being used.

    0 - Secondary.
    1 - Primary.

PD'FILLING(11).(5:1)

Indicates if the Initiation Manager is currently active and filling a write TBUF.

    0 - Not filling.
    1 - Filling. This is cleared after the TBUF has been filled
        and linked into the tail of write TBUFs.

PD'SSBRKENAB(11).(6:1)

Indicates if subsystem break is enabled.

    0 - Subsystem break is disabled. Subsystem break is
        disabled with an IOQ function code of 12 (FCONTROL 16)
        by the Initiation Manager.
    1 - Subsystem break is enabled. Subsystem break is enabled
        with an IOQ function code of 13 (FCONTROL 17) by the
        Initiation Manager.

PD'CONSENAB(11).(7:1)

Indicates if the console interrupt is enabled.

    0 - Console interrupt is disabled. This is disabled with
        an IOQ function code of 38 by the Initiation Manager.
    1 - Console interrupt is enabled. This is enabled with an
        IOQ function code of 38 by the Initiation Manager.

PD'BREAKENAB(11).(8:1)

Indicates if break is enabled.

    0 - Break is disabled. Break is disabled with an IOQ function
        code of 10 (FCONTROL 14) by the Initiation Manager.
    1 - Break is enabled. Break is enabled with an IOQ function
        code of 11 (FCONTROL 15) by the Initiation Manager.

PD'CRITICALW(11).(9:1)

Set by the Initiation Manager if the write is a critical write. At the com-
pletion of the write the logical device monitor is awakened and cleared by
the Initiation Manager.

PD'WAITFORTBUF(11).(10:1)

Indicates if the driver is currently waiting for a TBUF.

    0 - Not waiting for a TBUF.
    1 - Driver is waiting for a TBUF. This is set when the
        driver requests a TBUF and one is not available. See
        PD'DEVLINK for a complete explanation of when and how
        the driver gets TBUFs that are not available.

PD'PCC'XON'XOFF(11).(11:1)

Indicates if the hardware will do the XON/XOFF handshake.

    0 - The P&D driver will handle the XON/XOFF handshake.
    1 - The hardware will handle the XON/XOFF handshake.

WORD 12

PD'READTIME

Contains the computed read time, in 1/100ths of a second of the last timed
read. This is returned to the caller with an FCONTROL 18.

WORD 13

PD'TIMINGREAD(13).(0:1)

Indicates if the next read time is to be calculated.

    0 - Don't compute read time for next read. Cleared with an
        IOQ function code of 16 by the Initiation Manager when
        the next read is started.
    1 - Compute read time for next read. Set with an IOQ function
        code of 17 by the Initiation Manager when the next
        read is started. The computed read time is placed in
        PD'READTIME by the Interrupt Manager.

PD'RDTIMEOUTVAL(13).(1:15)

Contains a read time out value in .1's of seconds if the next read is to be
timed. This is set with an FCONTROL 5 by the Initiation Manager when the
next read is started.

PD'PREADTIMING

A reference to the above read timing information.

WORD 12 & WORD 13

PD'RDSTARTTIME

For reads that are to return a computed read time this double word will con-
tain the read start time. This is initially set by the Initiation Manager
when the read is started. When the read is complete the Interrupt Manager
places the computed read time in PD'READTIME.

WORD 14

PD'BRKTBUF

Contains a TDS relative pointer to the head TBUF of the broken read data.

WORD 15

PD'BROKRCNT

Contains the byte read count of the broken read.

WORD 16

PD'DEVLINK

Contains a TDS relative pointer to the next device waiting for a TBUF. The
pointer is to the P&D variable DIT. When the driver can't get a TBUF it will
set PD'WAITFORTBUF, indicating it is waiting for a TBUF. In the TDS header
(TDS'WAITHEAD'P and TDS'WAITTAIL'P) there is a linked list of devices waiting
for TBUFs. Again the list points to P&D variable DITs. The driver will then
link its variable DIT to the tail of the list. As TBUFs become free,
RETURNLYNXBUF will give the TBUF to the device at the head of the list and
then awaken the driver. The new TBUF pointer is placed, by RETURNLYNXBUF in
PD'TBUFWAIT in the variable DIT.

WORD 17

PD'TBUFWAIT

Contains a TBUF given to the driver (by RETURNTBUF) that was waiting for a
TBUF. See PD'DEVLINK for a complete explanation of when and how the driver
gets TBUFs that are not available.

WORD 16 & WORD 17

PD'DISCADDR

Contains a disc address for spooled reads.

WORD 18

PD'XFERCNT

Contains the current count on the number of bytes read if currently reading.
This count is only updated when the read is interrupted (i.e., TBUF becomes
full or read special character) and is complete or has to be restarted.

PD'TBUFS'IN'USE

A count on the number of write TBUFs currently in use. Each time the
Initiation Manager fills a TBUF this count is incremented. As the TBUFs be-
come empty the Interrupt Manager will decrement this count. When the count
gets to 2 and there is still data to be tanked (PD'BTANKED <> -1) then the
Initiation Manager is awakened to resume tanking.

WORD 19

PD'ERROR

Contains an error coded for the current transfer. This is a communication
word used by the Interrupt Manager to pass information to the Initiation
Manager.

    1 - No system or terminal buffer available to restart the read.
    2 - Overrun error.
    3 - Framing error.
    4 - Unused.
    5 - Parity error.
    6 - Unused.
    7 - Modem error.
    8 - Unused.
    9 - Unused.
    10 - A type 2 EOR has been detected.

WORD 20

NOTE:

This is a write only word for the Interrupt Manager. The Initiation Manager
should not write to any (used or unused) fields as it may cause software
problems.

PD'SUSPLOPSTATE(21).(0:1)

Set if a logical operation has been suspended that will resume later.

PD'CNTRLX(20).(1:1)

Indicates if the Control-X response should be sent when a Control-X is
received.

    0 - Don't send the Control-X response. The Control-X response
        is disabled with an IOQ function code of 27 by the Initiation
        Manager.
    1 - Control-X response is enabled. The Control-X response is
        enabled with an IOQ function code of 26 by the Initiation
        Manager.

PD'SPDSENSE(20).(2:1)

Indicates if the device has been speedsensed.

    0 - The device has not been speedsensed.
    1 - Set by the Interrupt Manager after a successful
        speedsense.

PD'ESCPAIR(20).(3:1)

Indicates if the driver is in the middle of processing an ESC sequence to maybe enable or disable echo.

    0 - Not processing an ESC pair sequence.
    1 - Driver is currently processing an ESC pair. Set by the
        Interrupt Manager when an "ESC" special character is
        detected. A one-byte read is then started to determine
        if echo should be changed.

PD'DC2READ(20).(4:1)

Indicates if a DC2 was read during the current read.

    0 - DC2 has not been read.
    1 - DC2 has been read. Set by the Interrupt Manager and
        indicates that the read is a View read, blockmode or
        lineblock mode read.

PD'XONWAIT(20).(5:1)

Indicates if the driver has read a DC3 (XOFF) and is waiting for the DC1 (XON). Note that this is only used when the driver processes the XON/XOFF. The hardware may actually do the XON/XOFF processing and the device may be in an XOFF state and this bit may not be set.

    0 - XOFF has not been read.
    1 - XOFF has been read.

PD'LBLOCKMODE(20).(6:1)

Indicates if the current read is a line block mode read.

    0 - The current read is not a line block mode read.
    1 - A DC2,CR has been read and the read is a line block
        mode read.

PD'BLOCKMODE(20).(7:1)

Indicates if the current read is a block mode read.

    0 - The current read is not a block mode read.
    1 - A DC2 has been read and the read is a block mode read.

PD'PENDLOPSTATE(20).(9:4)

This field contains the old logical operation. This is used when an event occurs (i.e., modem interrupt) that results in a new sequence or logical operations before the old one can be resumed.

PD'26318'RESET(20).(13:1)

Indicates if the HP 26318 is initially being reset. This is used to prevent bad status reports between FOPEN's, etc., that may occur when the printer is being reset.

    0 - HP 26318 not being reset.
    1 - HP 26318 being reset. Don't report any transfer errors.

PD'BINARY'MODE(20).(14:1)

Indicates if in binary mode.

    0 - Not in binary mode.
    1 - Binary mode enabled.

PD'LOGONDEV(20).(15:1)

Indicates if the device is a logon/speedsense device.

    0 - The device is not a logon device.
    1 - The device is a logon/speedsense device.

WORD 21

NOTE:

This is a write only word for the Initiation Manager. The Interrupt Manager should not write to any (used or unused) fields as it may cause software problems.

PD'READTYPE(21).(0:4)

This indicates the type of read that is currently active. This is set by the Initiation Manager and not changed until the read is logically complete. Different read types are:

    0 - No operation.
    1 - Character mode/block mode read.
    2 - Spooled read.
    3 - Idle read.
    4 - Transparent character/block mode read.
    5 - View/3000 read.
    6 - Binary read.
    7 - Not used.

PD'LLDMC(21).(4:3)

Contains the last logical device monitor call to the Initiation Manager.

    0 - Unused.
    1 - Set device characteristics (IM'SET'DEV).
    2 - Read (IM'READ).
    3 - Write (IM'WRITE).
    4 - Abort call (IM'ABORT).
    5 - Refuse call (IM'REFUSE).
    6 - Device control call (IM'DEV'CONTROL).
    7 - Unused.

PD'PRINTER(21).(7:1)

Set if logical device is a printer.

PD'READLOC(21).(8:2)

Indicates where the data is going for the current read.

    0 - Unused.
    1 - TBUFs are being used for current read.
    2 - System buffers are being used for current read.
    3 - Read data is going to a frozen data segment.

PD'LDMOPCODE(21).(10:6)

Contains the opcode of the last call by the monitor to the Initiation Manager. PD'LLDMC will indicate the last Initiation Manager call.

Opcodes for calls to IM'SET'DEV:

    0 - Partial completion. Initiation Manager should
        continue with old operation.
    1 - Return device characteristics.
    2 - Set a new term type.
    3 - Change parity.
    4 - Change the echo flag.
    5 - Change transparent read special characters.
    6 - Enable/disable subsystem break.
    7 - Enable/disable break.
    8 - Enable/disable Control-A.
    9 - Set/clear console mode.
    10- Set/clear console mode.
    11- Set data length.
    12- Disconnect.
    13- Enable/disable Control-X reply.
    14- Hangup timeout.
    15- Selftest.
    6 - Wait for current operation to complete.
    7 - Flush broken read TBUFs.
    8 - Disconnect immediately.

Opcodes for calls to IM'READ:

    0 - Continue current operation.
    1 - TBUF read.
    2 - Frozen stack read. Not used.
    3 - Spooled read.
    4 - Speedsense request.
    5 - Read timeout.
    6 - New read request.
    7 - Start pending read.
    8 - Break accepted.
    9 - Subsystem break accepted.

Opcodes for calls to IM'WRITE:

    0 - Continue current operation.
    1 - Tank data.
    2 - Tank token.
    3 - Start write.
    4 - HP 26318 status request check.

Opcodes for calls to IM'ABORT:

    0 - Unused.
    1 - Halt all I/O.
    2 - Abort current operation and start idle read.
    3 - Hard preempt.
    4 - Soft preempt.
    5 - Break accepted.
    6 - Subsystem break accepted.

Opcodes for calls to IM'REFUSE:

    0 - Unused.
    1 - Break refused.
    2 - Subsystem break refused.

WORD 22

PD'STATUS

Contains the status byte returned from an HP 26318 printer.

    Bit 4 - If set then a transfer error occurred.
    Bit 5 - If set then the device is offline.
    Bit 6 - If set then the device buffer is full.
    Bit 7 - If set then the device is out of paper.

WORD 23

PD'OLDXFERCNT

Indicates the number of bytes read up to the last back space. Needed so the successive LF's are not output on devices that required a LF in response to a BS. Set and cleared by the Interrupt Manager.

PD'STATUS'WCNT

Contains the write count for a HP 2631B status request.

WORD 24

PD'EOFTBUF

Contains a TDS relative pointer to the saved EOF TBUF(s).

WORD 25

PD'EOFCNT

Contains a byte count of saved EOF data in PD'EOFTBUF.

WORD 26

PD'READTIMERINDEX

A timer index for reads that are to be timed.

Port Protocol DIT Format

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
0																	
13	TERMTYPE FILE NAME																11
14	TERMINAL DST RELATIVE POINTER TO VFC ENTRY																12
15	ECH	ENQ	DLY		XFL	XST	EMS	CST						FOK	DC3	BLKMODE	13
16	XON	WST	31B	INI	VFC	CHAR SIZE			NEW FORM FEED CHARACTER								14
17	WRM	CR DELAY				LF DELAY				FF DELAY							15
20	ENQ/ACK BLOCK SIZE								ENQUIRY CHARACTER								16
21				ACK OPTION			ACKNOWLEDGE CHARACTER										17
22	BLOCK MODE READ TRIGGER CHAR								READ TRIGGER CHARACTER								18
23															BS ACTION		19
24																	20
35	BLOCK MODE CURSOR STRING																29
36																	30
42																	34
43	LAST SSBREAK CHARACTER							RESERVED				LOGON PARITY				35
44	RESERVED				ODD PARITY			RESERVED				EVEN PARITY				36
45								XON TIME VALUE IN SECONDS								37
46																	38
51																	41
52	SCFMA - 0				SCFMA - 1			SCFMA - 2				SCFMA - 3				42
111	SCFMA - 124				SCFMA - 125			SCFMA - 126				SCFMA - 127				73
112	0																74
113	1																75

WORDS 0/11

Contains a file name of the termtype file.

WORD 12

PP'VFC

Terminal DST relative pointer to a VFC entry in the VFC table.

WORD 13

PP'ECHO.(0:1)

Indicates the initial state of echo.

    0 - Echo is disabled.
    1 - Echo is enabled.

PP'ENQACK.(1:1)

Indicates if the device does ENQ/ACK handshaking.

    0 - ENQ/ACK handshaking is disabled.
    1 - ENQ/ACK handshaking is enabled. PP'ENQCHAR and
        PP'ACKCHAR will contain the ENQ/ACK character and
        PP'ENQBLOK will contain the block size.

PP'DELAY.(2:1)

Indicates if delays should be enabled.

    0 - Delays are not enabled.
    1 - Delays are enabled. PP'DELAYCR, PP'DELAYLF and
        PP'DELAYFF will contain the delay amount in 10ths of
        seconds.

PP'XFLOW.(4:1)

Indicates if XON/XOFF handshaking should be enabled.

    0 - XON/XOFF handshaking is disabled.
    1 - XON/XOFF handshaking is enabled.

PP'XSTRIP.(5:1)

Indicates if the XON/XOFF should be stripped.

    0 - The XON/XOFF should not be stripped from read data.
    1 - The XON/XOFF will be stripped from read data if
        handshakes are disabled.

PP'EMSTRIP.(6:1)

Indicates if Control-Y should be stripped.

    0 - Control-Y should not be stripped from read data.
    1 - Control-Y will be stripped from read data.

PP'CONS'STRIP.(7:1)

Indicates if Control-A should be stripped.

    0 - Control-A should not be stripped from read data.
    1 - Control-A will be stripped from read data.

PP'FFOK.(12:1)

Indicates if FF should be allowed as output.

    0 - The FF should be replaced by PP'FF'NEWCHAR.
    1 - The FF are valid.

PP'DC3'CONTROL.(13:1)

Indicates if a DC3 should be appended to write data after each CF, LF.

    0 - Don't append any DC3s to write data.
    1 - Append a DC3 following the CF, LF.

PP'BLOCKMODE.(14:2)

Indicates the type of blockmode read to do on DC2s.

    0 - None.
    1 - Line blockmode.
    2 - Page blockmode.
    3 - Either line or page blockmode.

WORD 14

PP'DO'XON'TIMER.(0:1)

Indicates if the driver should start an XON timer after an XOFF is received.

    0 - Don't start an XON timer if an XOFF is received.
    1 - Start an XON timer, indicated by PP'XON'TIME
        if an XOFF is received.

PP'WRITESTATUS.(1:1)

Indicates if a status request, "ESC?DC1", should be written to the device.

    0 - Don't request status from the device.
    1 - Send a status request to the device.

PP'2631B'FIX.(2:1)

Indicates if status should be requested from the device after an XOFF even though PP'WRITESTATUS may be set.

    0 - Don't worry about when the status request is sent.
    1 - Don't request status as the XOFF was received in the
        middle of a line and we don't want to possibly send
        an ESC sequence in the middle of a user ESC sequence.

PP'INIT'DEV.(3:1)

Indicates if an initialization sequence should be sent to the device.

    0 - There is no initialization sequence.
    1 - There is an initialization sequence.

PP'VFC'OK.(4:1)

Indicates if there is a VFC file for the device.

    0 - There is no VFC file to send to the device.
    1 - There is a VFC file for the device.

PP'CHAR'SIZE.(5:3)

Indicates the size of the data characters. The value is 1 less than the actual character size, i.e., 8-bit data will be indicated by a 7.

PP'FF'NEWCHAR.(8:8)

Contains the replacement character for FFs if FFs are to be replaced.

WORD 15

PP'NAME'VALID.(0:1)

Indicates if the term type file name is the current term type.

    0 - The current term type is specified by a numbered
        term type. PD'TERMTYPE will contain a number and the
        file being used will be TERM"number".PUB.SYS.
    1 - The current term type is specified by a file name.
        PD'TERMTYPE will contain a 0.

PP'DELAYCR.(1:5)

Indicates the amount of time in 10ths of seconds to delay on CR.

PP'DELAYLF.(6:5)

Indicates the amount of time in 10ths of seconds to delay on LFs.

G.23.00
25- 37

PP'DELAYFF.(11:5)

Indicates the amount of time in 10ths of seconds to delay on FFs.

WORD 16

PP'ENQBLOK.(0:8)

Indicates the number of characters to send before doing the ENQ/ACK handshake.

PP'ENQCHAR.(8:8)

Contains the inquire character, normally the ENQ.

WORD 17

PP'NOACKACTION.(5:3)

Indicates what to do if the ACK is not received on an ENQ/ACK handshake.

    1 - Resume write with no ENQ.
    2 - Resume write with ENQ.

PP'ACKCHAR.(8:8)

Contains the acknowledge character, normally the ACK.

WORD 18

PP'BLOCK'TRIG.(0:8)

Contains the blockmode read trigger character, normally a DC1.

PP'TRIGGER'CHAR.(8:8)

Contains the read trigger character, normally a DC1.

WORD 19

PP'BSRESP.(13:3)

Indicates the response for the back space character.

    1 - Nothing.
    2 - Send end of medium.
    3 - Send a "LF".
    4 - Send a "/".
    5 - Erase the character.

G.23.00
25- 38

WORDS 20/29

Contains the block mode cursor string.

WORD 35

PP'LAST'SSBRK.(0:8)

Contains the last subsystem break character detected.

PP'PARITY'ENAB.(12:1)

Indicates if parity checking should be enabled when the device is FOPEN'ed.

    0 - Parity checking is disabled.
    1 - Parity checking is enabled.

PP'FOPEN'PARITY.(13:3)

Indicates what the parity should be when the device is FOPEN'ed.

    0 - Space.
    1 - Mark.
    2 - Even.
    3 - Odd.

WORD 36

PP'ODD'ENAB.(4:1)

Indicates if parity checking should be enabled if odd parity is sensed.

    0 - Parity checking is disabled.
    1 - Parity checking is enabled.

PP'ODD'PARITY.(5:3)

Indicates what the parity should be if odd (0) parity is sensed.

    0 - Space.
    1 - Mark.
    2 - Even.
    3 - Odd.

PP'EVEN'ENAB.(12:1)

Indicates if parity checking should be enabled if even parity is sensed.

    0 - Parity checking is disabled.
    1 - Parity checking is enabled.

G.23.00
25- 39

PP'EVEN'PARITY.(13:3)

Indicates what the parity should be if even (1) parity is sensed.

    0 - Space.
    1 - Mark.
    2 - Even.
    3 - Odd.

WORD 37

PP'XON'TIME.(8:8)

Contains the amount of time in seconds to wait for the XON after an XOFF.

WORDS 42/73

Contains a special character function code for each of the 128 ASCII characters. This is an array of 128 4-bit entries. There is one entry for each character, going sequentially from 0 to 127. The special character function codes are as follows:

    0 - No special function.
    1 - Console attention (i.e., Control-A).
    2 - Cancel one character (i.e., backspace).
    3 - Horizontal tab.
    4 - Linefeed.
    5 - Type 1 end-of-record (i.e., CR).
    6 - Type 2 end-of-record (i.e., an IOQ EOR character).
    7 - XON
    8 - Block mode alert character (i.e., DC2).
    9 - XOFF
    10- Rewrite input buffer.
    11- Cancel line (i.e., Control-X).
    12- Subsystem break (i.e., Control-Y).
    13- Strip and ignore.
    14- Escape character.

G.23.00
25- 40

Hardware DIT Format

ATP/ATP37 Hardware DIT Format

```
       0  1  2  3  4  5  6  7  8  9  10 11 12 13 14 15
      |---|--|--|--|--|--|--|--|--|--|---|---|---|---|---|---|
    0| SYSDB RELATIVE POINTER TO LOGICAL DEVICE MONITOR DIT  |
      |------------------------------------------------------|
    1| TERMINAL DST RELATIVE POINTER TO PROTOCOL & DATA DIT  |
      |------------------------------------------------------|
    2| TERMINAL DST RELATIVE POINTER TO CONTROL PROGRAM AREA |
      |---------|---|---|-----------|------------------------|
      |CONTROL|MOD|NON| WAIT        |                        |
    3| TYPE   |CON|55 | REASON      | TRUE HARDWARE UNIT NUMBER|
      |---|---|---|---|---|---|-----|------------------------|
      |PRI|ECO|IDL|SPD|SET|         |                        |
    4|CHR|ENA|WRT|SPC|WRK| LAST HW STATE |    HARDWARE STATE |
      |---|---|---|---|---|---|---|-------|------------------|
    5|SPP|PRF|   |EO2|BRO|TTY| FF|XON|WID|PAR|GEN| CK| LINE SPEED|
      |---|---|---|---|---|---|---|---|---|---|---|---|---------|
    6| NEXT TO LAST INTERRUPT REASON | LAST INTERRUPT REASON |
      |-------------------------------|-----------------------|
    7| READ COUNT                                            |
      |------------------------------------------------------|
   10|***|R/L| READ BANK                                  |8
      |---|---|----------------------------------------------|
   11| READ ADDRESS                                       |9
      |------------------------------------------------------|
   12|***|R/L| WRITE BANK                                 |10
      |---|---|----------------------------------------------|
   13| WRITE ADDRESS                                      |11
      |------------------------------------------------------|
   14| WRITE COUNT                                        |12
      |------------------|-------|------------|-------|-------|
   15| OLD DIRECT CMD   | DIRECT COMMAND    |  OLD WAIT  |13
      |------------------|-------------------|-------------|
   16| FRAMING ERROR COUNT   | LAST SPECIAL CHARACTER     |14
      |---|---------------|---|----------------|---|-------|
   17|P/F| PCC DATE CODE |P/F| MCC DATE CODE |P/F| MSC DATE CODE |JPT|15
      |---|---------------|---|--------------|---|---------------|---|
   20| NEXT STATE     | PS| MODEM OUTPUT CONTROL          |16
      |----------------|---|-------------------------------|
   21| MODEM INPUT REFERENCE  | MODEM INPUT CONTROL       |17
      |------------------------|----------------------------|
   22|                                                   |18
   31| PRIMARY SPECIAL CHARACTER MASK (8 WORDS)          |25
      |------------------------------------------------------|
```

ATP/ATP37 Hardware DIT Format (Cont.)

```
      |------------------------------------------------------|
   32|                                                   |26
   35| SECONDARY SPECIAL CHARACTER SET (4 WORDS)         |29
      |------------------------------------------------------|
   36|                                                   |30
   41| WRITE SPECIAL CHARACTER SET (4 WORDS)             |33
      |------------------------------------------------------|
   42|                                                   |34
   45| WRITE EDIT SPECIAL CHARACTER SET (4 WORDS)        |37
      |------------------------------------------------------|
   46| WRITE BUFFER                                      |38
      |------------------------------------------------------|
   47|                                                   |39
   56| READ BUFFER (8 WORDS)                             |46
      |---|---|----------------------------------------------|
   57|DIR|DIN|             DIAGNOSTIC INTERRUPT CODE     |47
      |---|---|----------------------------------------------|
   60|                                                   |48
   61| SAVED CLOCK VALUE - (2 WORDS)                     |49
      |-----------|-----------|--------------|------------|
   62| BLOCK COUNT | CR DELAY | ENQ CHARACTER | LF DELAY  |50
      |-------------|----------|--------------|------------|
   63| ACK CHARACTER | FF DELAY |             |           |51
      |---------------|----------|--------------|----------|
```

WORD 0

HW'LDITP

SYSDB relative pointer to the logical monitor DIT.

WORD 1

HW'PDITP

Terminal DST relative pointer to the protocol and data management DIT.

WORD 2

HW'CP'P

Terminal DST relative pointer to the control program area. Control program area is 32 bytes.

WORD 3

HW'CONTROLLER(3).(0:2)

Indicates the type of controller.

    0 - Unused.
    1 - ATP controller.
    2 - RDCC controller.
    3 - Unused.

HW'MODEMPANEL(3).(2:1)

Indicates the type of connection/junction panel.

    0 - Direct connection, device subtype was configured as 0 or 14.
    1 - Modem connection, device subtype was configured as 1 or 15.

HW'NON55(3).(3:1)

Indicates the type of CPU.

    0 - Series 64 type CPU.
    1 - Series 40/44 type CPU.

HW'WAIT'REASON(3).(4:4)

This indicates how the next interrupt should be processed. A wait reason is set up when the driver is about to halt the PCC and will want to selectively process the next interrupt. Wait reasons are as follows:

    0 - No wait reason. Process the interrupt as per the interrupt
        type.
    1 - Abort pending. The physical driver is trying to halt the
        PCC. Modem interrupts are processed otherwise all other
        interrupts satisfy the halt.
    2 - Reset DIT. The port is being reset. When the next
        interrupt occurs the reset will be completed.
    3 - Disconnect. The PCC is being halted so that a control
        program can be started to disconnect the modem. Any
        interrupt will satisfy the halt.
    4 - MCC setup. The MCC is being set up and the driver is
        waiting for the "end of control program" interrupt
        indicating the MCC has been set up. Any interrupt except
        modem errors will satisfy the setup complete.
    5 - Monitor modem signals. The driver has set up the MCC and
        is waiting for an interrupt indicating that a modem line
        is in the correct state, i.e., DSR on.
    6 - EO2 reset. The PCC is hung and the driver is in the
        process of resetting the PCC before it does the next read or
        write.

---

    7 - Dump port. The PCC is dumping its RAM.
    8 - Speed-specified. The hardware is waiting for the "end
        of control program" interrupt indicating that the hardware
        has been set up accordingly as a speed-specified port.

HW'UNIT'NUM(3).(8:8)

Indicates the true unit number of the device. Unit numbers will range from 0 to 127.

WORD 4

HW'PRISPCL(4).(0:1)

Indicates what read special character set is being used.

    0 - Secondary read special character set enabled.
    1 - Primary read special character set enabled.

HW'ECHO(4).(1:1)

Indicates the current state of echo.

    0 - Echo is disabled.
    1 - Echo is enabled.

HW'IDLE'WRT(4).(2:1)

Indicates if the current perform I/O will write out one or two trigger characters before the read begins.

    0 - No trigger characters are to be sent.
    1 - One or two trigger characters are to be written.

HW'SPDS(4).(3:1)

Indicates whether the port is a speed-specified port.

    0 - The port is speedsensing.
    1 - The port is speed-specified.

HW'MCC'SETUP'WRK(4).(4:1)

Indicates whether we wait for end CP after setting up modem signals.

    0 - Wait for end CP.
    1 - Do not wait for end CP.

HW'LSTSTATE(4).(6:4)

This indicates the last state of the PCC. When an interrupt occurs the current state in HW'STATE is saved in HW'OLD'STATE. Then HW'STATE is set to "input save".

HW'STATE(4).(12:4)

This indicates the current state of the PCC. Set when a PCC control program is started or after an interrupt occurs. Possible states are as follows:

```
1  - Reading.
2  - Writing.
3  - Speedsensing.
4  - Unused.
5  - Set port protocol.
6  - Set special characters.
7  - Selftest.
8  - Dumping PCC.
9  - Port is frozen.
10 - Reading modem inputs.
11 - Reset.
12 - Idle read.
13 - Setting up modem signals.
14 - Monitoring modem signals.
15 - Input save.
```

WORD 5

HW'SET'PROTOCOL(5).(0:1)

Indicates if the PCC has done a set port protocol.

```
0 - The PCC has not done a set port protocol. No I/O
    should be done until after a set port protocol.
1 - The PCC has done a set port protocol. The current
    protocol is indicated in bits 11/15 of this word.
```

HW'POWERFAIL(5).(1:1)

Indicates if a power fail has occurred and is being processed.

```
0 - No power fail has occurred.
1 - Power fail has occurred.
```

HW'RESET'DIT(5).(2:1)

Indicates if the hardware DIT is being reset.

```
0 - DIT is not being reset.
1 - DIT is being reset.
```

HW'CLEAR'EO2(5).(3:1)

Indicates if the driver is in the process of resetting the PCC to clear up the problem of the PCC getting hung because location EO of its RAM never is cleared.

```
0 - The driver is not trying to fix a hung PCC.
1 - The driver is in the middle of trying to free a hung PCC.
```

HW'BROKEN(5).(4:1)

Indicates if the port is broken.

```
0 - The port is not broken.
1 - The port is broken and will not operate until reset.
```

HW'DELAY'ENAB(5).(5:1)

Indicates if TTY delays are enabled.

```
0 - TTY delays are disabled.
1 - TTY delays are enabled. There will be a delay
    following the transmission of each "CR", "LF", or "FF".
```

HW'FF'ENAB(5).(6:1)

Indicates if form feeds are enabled.

```
0 - Form feeds are disabled. Each form feed character will
    be replaced with a "LF".
1 - Form feeds are enabled. Each "FF" character is written
    out and not replaced with a "LF".
```

HW'XON'ENAB(5).(7:1)

Indicates if the PCC will do the XON/XOFF handshake.

```
0 - The PCC will not do the XON/XOFF handshake.
1 - The PCC will do the XON/XOFF handshake.
```

HW'8'BIT'MODE(5).(8:1)

Indicates the size of the data character.

```
0 - Data will be transmitted as 7-bit with a parity bit.
1 - Data will be transmitted as 8-bit data.
```

HW'PARITY'GEN(5).(9:2)

Indicates the type of parity generation. This field is only valid for 7-bit data.

```
0 - Output disabled and the 8th bit is forced to 0.
1 - Output disabled and the 8th bit is forced to 1.
2 - Output parity generation is enabled and it will be even.
3 - Output parity generation is enabled and it will be odd.
```

HW'PARITY'CHECK(5).(11:1)

Indicates if input parity checking is enabled.

```
0 - Input parity checking is disabled.
1 - Input parity checking is enabled.
```

HW'LINE'SPEED(5).(12:4)

Indicates the current transfer rate of the PCC.

```
0  -   110 baud.
1  -   300 baud.
2  -   600 baud.
3  -  1200 baud.
4  -  2400 baud.
5  -  4800 baud.
6  - 19200 baud.
7  -  9600 baud.
8  - 76800 baud - unsupported.
9  -  9600 baud.
10 -  1200 baud.
11 -   300 baud.
```

WORD 6

HW'LAST'INTERRUPT(6).(0:8)

Contains the old interrupt code. When an interrupt occurs the interrupt code in HW'INTERRUPT'CODE is moved here and the new interrupt is placed in HW'INTERRUPT'CODE.

HW'INTERRUPT'CODE(6).(8:8)

Contains the last interrupt code. Some of the interrupt codes have different meanings depending on the PCC version. Those different codes are noted. Interrupt codes 20/28 are only for pass 3 PCCs.

```
0  - Invalid.
1  - Redundant start I/O.
2  - End control program.
3  - Illegal control program.
4  - Write special character.
5  - Ten-second ACK timeout.
6  - Read complete.
7  - Parity error.
8  - Read special character.
9  - Pass 2 PCCs - break detected. Pass 3 PCCs - framing error.
10 - Overrun error.
11 - Pass 2 PCCs - character read and not in read state.
     pass 3 PCCs - not used.
12 - Speedsense complete.
13 - Pass 2 PCCs - modem error. Pass 3 PCCs - break detected.
14 - Write complete.
15 - Selftest complete.
16 - Edit special character.
17 - Diagnostic failure.
18 - Control program halted.
19 - Dump complete.
20 - Modem data overrun.
21 - Modem RIB hardware error.
22 - Modem invalid PCC read.
23 - Modem PCC message error.
24 - Modem link error.
25 - Modem debounce error.
26 - Modem non-maskable interrupt error.
27 - MC6801 error.
28 - Unknown modem error.
```

WORD 7

HW'READ'CNT

Current PCC read byte count.

WORD 8

Note that this word is used in full for the ATP DMA registers. It should not be changed in its format.

HW'RD'RIGHT'LEFT(8).(1:1)

Indicates to the PCC what byte to start the transfer to.

    0 - Left.
    1 - Right.

HW'READ'BANK

Indicates the bank number for the current read.

WORD 9

HW'READ'ADDR

Contains the absolute start address of the read.

WORDS 8/9

HW'READ'ABS'ADDR

Contains a double word absolute start address for the read.

WORD 10

Note that this word is used in full for the ATP DMA registers. It should not be changed in its format.

HW'WT'RIGHT'LEFT(10).(1:1)

Indicates to the PCC what byte to start the write from.

    0 - Left.
    1 - Right.

HW'WRITE'BANK

Indicates the bank number for the current write.

WORD 11

HW'WRITE'ADDR

Contains the absolute start address of the write.

WORDS 10/11

HW'WRITE'ABS'ADDR

Contains a double word absolute start address of the write.

WORD 12

HW'WRITE'CNT

Current PCC write byte count.

WORD 13

HW'OLD'DIRECT'CMD(13).(0:4)

Contains the old direct command. When a new direct command is issued HW'DIRECT'COMMAND is saved here. Then the new direct command is saved in HW'DIRECT'COMMAND. Possible direct commands are:

    1 - Modem freeze.
    2 - PCC freeze.
    4 - Start I/O.
    8 - Halt direct command.

HW'DIRECT'COMMAND(13).(6:4)

Contains the last direct command issued. When a new command is issued the old command is saved in HW'OLD'DIRECT'CMD.

HW'OLD'WAIT'REASON(13).(13:3)

Contains the old wait reason. HW'WAIT'REASON is saved here before a new wait reason is set up.

WORD 14

HW'FRAMING'ERROR(14).(0:8)

Contains a running count on the number of framing errors that occur. Used only for gathering information when dumps are read.

HW'SPEC'CHAR(14).(8:8)

Contains the last special character detected. Updated each time a new special character is detected.

WORD 15

HW'SELFTEST

Contains the results from the PCC/MSC selftest.

HW'PCC'PF(15).(0:1)

Indicates the results of the PCC selftest.

    0 - Selftest passed.
    1 - Selftest failed.

HW'PCC'DATE(15).(1:4)

Contains the current PCC date code.

    0 - Pass 1 PCC.
    1 - Pass 2 PCC.
    2 - Pass 3 PCC.
    3 - Pass 4 PCC.

HW'MCC'PF(15).(5:1)

Indicates the results of the MCC selftest.

    0 - Selftest passed.
    1 - Selftest failed.

HW'MCC'DATE(15).(6:4)

Contains the current MCC date code.

    0 - Pass 1 MCC.
    1 - Pass 2 MCC.
    2 - Pass 3 MCC.

HW'MSC'PF(15).(10:1)

Indicates the results of the MSC selftest.

    0 - Selftest passed.
    1 - Selftest failed.

HW'MSC'DATE(15).(11:4)

Contains the current MSC date code.

    0 - Pass 1 MSC.
    1 - Pass 2 MSC.
    2 - Pass 3 MSC.

HW'JUNCTION'TYPE(15).(15:1)

Indicates the type of junction panel.

WORD 16

HW'NEXT'STATE(16).(0:4)

Indicates the next state of the PCC. See HW'PENDING'START for an explanation of how this field is used.

HW'PENDING'START(16).(6:1)

Indicates if there is a pending start. This occurs when the driver is active and a modem signal changes preventing the next read or write from starting. Everything is saved in the DIT and when the modem line is correct (i.e., DCD is on) the read or write will be started.

    0 - No pending start.
    1 - A pending read or write is ready.

HW'MODEM'OUTPUT(16).(8:8)

Contains an 8-bit modem output control mask. If a bit is set then that signal will be "on" or plus 12 volts.

    Bit 0 - Frequency select.
    Bit 1 - High order binary digit.
    Bit 2 - Second order binary digit.
    Bit 3 - Low order binary digit.
    Bit 4 - Secondary request to send.
    Bit 5 - Call request.
    Bit 6 - Request to sent.
    Bit 7 - Data terminal ready.

WORD 17

HW'MODEM'REF(17).(0:8)

Contains an 8-bit modem reference mask. There is one bit for each input signal. If the signal is different from the reference and is a needed signal as specified by the control mask then an interrupt will occur.

    Bit 0 - Unused.
    Bit 1 - Clear to send.
    Bit 2 - Signal quality.
    Bit 3 - Data set ready.
    Bit 4 - Call origin status.
    Bit 5 - Secondary carrier detect.
    Bit 6 - Ring indicator.
    Bit 7 - Carrier detect.

HW'MODEM'CTL(17).(8:8)

Contains an 8-bit modem control mask. There is one bit for each input signal. If the bit is set then the signal will be used, otherwise it is ignored. The mask is the same as in HW'MODEM'REF.

WORDS 18/25

HW'PRI'SCHRS

Contains an 128-bit mask of the primary special character set. There is 1-bit for each of the 128 ASCII characters. If set then the character is a special character. The bit map has to be looked at as a 16-byte map where the characters are numbered 0-7, right to left. For example if Control-A is a special character the first word of the table would be X1000. If backspace was set the first word would be 1.

WORDS 26/29

HW'SEC'SCHRS

Contains an 8-character buffer for a secondary special character set.

WORDS 30/33

HW'WRI'SCHRS

Contains an 8-character buffer for write special characters.

WORDS 34/37

HW'EDIT'SCHRS

Contains an 8-character buffer for write edit special character set.

WORD 38

HW'WRITE'BUFR

Contains a 2-character write buffer used to send trigger characters for reads.

WORDS 39/46

HW'READ'BUFFER

Contains a 16-character read buffer used by the PCC for idle reads.

WORD 47

HW'DIAGNOSTIC(47).(0:1)

Indicates if the diagnostics are running.

0 - The diagnostics are not running.
1 - The diagnostics are running and all interrupts will be processed by the diagnostics.

HW'DIAG'INTERRUPT(47).(1:1)

Indicates if an interrupt occurred that has to be processed by the diagnostics.

0 - No interrupt has occurred.
1 - An interrupt has occurred and the interrupt reason is in HW'DIAG'REASON.

HW'DIAG'REASON(47).(8:8)

Contains the current diagnostic interrupt code. The current interrupt code is saved here when an interrupt occurs and the diagnostic bit is set. The interrupt codes are the same as specified in HW'INTERRUPT'CODE.

WORD 48/49

HW'MODEM'TIME

Contains the current clock value when a modem control program is started. When the driver is to halt the PCC and this word is non-zero then the halt will not occur until 200 msec have passed. This is to prevent a modem over-run error.

WORD 50

HW'PP1(50).(0:8)

Contains either the ENQ/ACK block count or the CR delay.

HW'PP1(50).(8:8)

Contains either the ENQ character or the LF delay, depending on whether we use ENQ/ACK handshake or time delay for flow control.

WORD 51

HW'PP2(51).(0:8)

Contains either the ACK character or the FF delay.

ADCC Hardware DIT Format

```
    0  1  2  3  4  5  6  7  8  9  10 11 12 13 14 15
  |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
 0| SYSDB RELATIVE POINTER TO LOGICAL DEVICE MONITOR DIT |
  |-----------------------------------------------------|
 1| TERMINAL DST RELATIVE POINTER TO PROTOCOL & DATA DIT |
  |-----------------------------------------------------|
 2| SYSDB RELATIVE POINTER TO CHANNEL PROGRAM AREA       |
  |CONTROL|MOD|NON|          NINE BIT DRT NUMBER         |
 3|TYPE   |CON|55 |   IMB # | CHANNEL # | DEVICE #       |
  |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
  |PRI|XON|SET|   |   |              |                   |
 4|CHR|XOF|WRK|   |   |  LAST HW STATE | HARDWARE STATE  |
 5|SPP|PRF|BRK|   |BRO|TTY|FF|XON|WID|PAR|GEN|CK| LINE SPEED|
 6| NEXT TO LAST INTERRUPT REASON | LAST INTERRUPT REASON |
  | CPVA #|  INTERRUPT CODE  |CPVA #|  INTERRUPT CODE |
 7| READ COUNT                                        |
10|***|R/L| READ BANK                              |8
11| READ ADDRESS                                   |9
12|***|R/L| WRITE BANK                             |10
13| WRITE ADDRESS                                  |11
14| WRITE COUNT                                    |12
15| LAST CHAR TO NEED DELAYS | INT. NEEDED AFTER DELAY |13
16| FRAMING ERROR COUNT | LAST SPECIAL CHARACTER    |14
17| SPEEDSENSE STARTED TIMER REQUEST               |15
20| NEXT STATE |       MASK FOR ALL MODEM SIGNALS OFF|16
21| MODEM INPUT REFERENCE | MODEM INPUT CONTROL     |17
22|                                                |18
31| PRIMARY SPECIAL CHARACTER MASK (8 WORDS)       |25
32|                                                |26
35| SECONDARY SPECIAL CHARACTER SET (4 WORDS)      |29
```

ADCC Hardware DIT Format (Cont.)

```
 36|                                                |30
 41| WRITE SPECIAL CHARACTER SET (4 WORDS)          |33
 42|                                                |34
 45| WRITE EDIT SPECIAL CHARACTER SET (4 WORDS)     |37
 46| WRITE BUFFER                                   |38
 47|                                                |39
 56| READ BUFFER (8 WORDS)                          |46
 57|DIA|DIN|           DIAGNOSTIC INTERRUPT CODE    |47
 60| SPEED USED WHEN PORT IS SUBTYPE 4 OR 5         |48
 61| TRACE WORD FOR COUNTING INTERRUPTS             |49
 62| ENABLE XMIT NOT FULL MASK | DISABLE XMIT NOT FULL MASK |50
 63| CURRENT ECHO, MODEM SIGNALS | ECHO OFF, MODEM SIGNALS |51
 64| INPUT SAVE CHARACTER READ | INPUT SAVE STATUS READ |52
 65| SET UP UART PARAMETERS | INPUT SAVE MODEM SIGNALS READ |53
 66|OE|PE|FE|BRK|MDM|NA|                    |XON|PCK|54
 67| ENQUIRY BLOCK COUNT | CHARS LEFT TO ENQ/ CR PAD |55
 70| ENQUIRY CHARACTER/ LF PAD | ACKNOWLEDGE CHAR/ FF PAD |56
 71| ACKNOWLEDGE WAIT TIMER REQUEST                 |57
 72|                                                |58
111| HARDWARE SPECIAL CHARACTER BIT MAP   (16 WORDS) |73
112|                                                |74
121| INPUT SAVE BUFFER  (8 WORDS)                   |81
122|                                                |82
141| BUFFER SPACE USED TO CHANGE SPECIAL CHARACTER ROMS (16 WORDS) |97
```

WORD 0

HW'LDITP

SYSDB relative pointer to the logical monitor DIT.

WORD 1

HW'PDITP

Terminal DST relative pointer to the protocol and data management DIT.

WORD 2

HW'CP'P

SYSDB relative pointer to the channel program area.

WORD 3

HW'CONTROLLER(3).(0:2)

Indicates the type of controller.

    0 - Unused.
    1 - ATP controller.
    2 - ADCC controller.
    3 - Unused.

HW'MODEMPANEL(3).(2:1)

Indicates the type of connection/junction panel.

    0 - Direct connection, device subtype was configured as 0 or 14.
    1 - Modem connection, device subtype was configured as 1 or 15.

HW'NON55(3).(3:1)

Indicates the type of CPU.

    0 - Series 64 type CPU.
    1 - Series 40/44 type CPU.

HW'DRT(3).(7:9)

Contains the 9-bit DRT number of the device. This consists of a 2-bit IMB number, 4-bit channel number, and 3-bit device number.

WORD 4

HW'PRISPCL(4).(0:1)

Indicates what read special character set is being used.

    0 - Secondary read special character set enabled.
    1 - Primary read special character set enabled.

HW'DO'XON'XOFF(4).(1:1)

This bit indicates if XON/XOFF handshaking is enabled.

HW'SETUP'WAKE(4).(2:1)

Indicates whether we wait for end CP after setting up modem signals.

    0 - Wait for end CP.
    1 - Do not wait for end CP.

HW'OLD'STATE(4).(6:4)

This indicates the last state of the ADCC. When an interrupt occurs the current state in HW'STATE is saved in HW'OLD'STATE, if the state is not "input save". Then HW'STATE is set to "input save".

HW'STATE(4).(12:4)

This indicates the current state of the ADCC. Set when an ADCC channel program is started or after an interrupt occurs. Possible states are as follows:

    1 - Reading.
    2 - Writing.
    3 - Speedsensing.
    4 - Generating an interrupt from an input save event.
    5 - Set port protocol.
    6 - Set special characters.
    7 - Unused.
    8 - Unused.
    9 - Unused.
    10 - Unused.
    11 - Outputting trigger characters.
    12 - Idle read.
    13 - Setting up modem signals.
    14 - Monitoring modem signals.
    15 - Input save.

WORD 5

HW'SET'PROTOCOL(5).(0:1)

Indicates if the ADCC has done a set port protocol.

    0 - A set port protocol has not been done.
    1 - A set port protocol has been done. The current
        protocol is indicated in bits 11/15 of this word.

HW'POWERFAIL(5).(1:1)

Indicates if a power fail has occurred and is being processed.

    0 - No power fail has occurred.
    1 - Power fail has occurred.

HW'BREAK'DETECTED(5).(2:1)

Indicates that a BREAK condition has been detected.

    0 - No BREAK condition detected.
    1 - A BREAK condition is currently being processed.

HW'BROKEN(5).(4:1)

Indicates if the port is broken.

    0 - The port is not broken.
    1 - The port is broken and will not operate until reset.

HW'DELAY'ENAB(5).(5:1)

Indicates if TTY delays are enabled.

    0 - TTY delays are disabled.
    1 - TTY delays are enabled. There will be a delay
        following the transmission of each "CR", "LF", or "FF".

HW'FF'ENAB(5).(6:1)

Indicates if form feeds are enabled.

    0 - Form feeds are disabled. Each form feed character will
        be replaced with a "LF".
    1 - Form feeds are enabled. Each "FF" character is written
        out and not replaced with a "LF".

HW'XON'ENAB(5).(7:1)

Indicates if the driver will do the XON/XOFF handshake.

    0 - The driver will not do the XON/XOFF handshake.
    1 - The driver will do the XON/XOFF handshake.

HW'8'BIT'MODE(5).(8:1)

Indicates the size of the data character.

    0 - Data will be transmitted as 7-bit with a parity bit.
    1 - Data will be transmitted as 8-bit data.

HW'PARITY'GEN(5).(9:2)

Indicates the type of parity generation. This field is only valid for 7-bit data.

    0 - Output parity generation is enabled and it will be even.
    1 - Output parity generation is enabled and it will be odd.
    2 - Output parity generation is enabled and it will be even.
    3 - Output parity generation is enabled and it will be odd.

HW'PARITY'CHECK(5).(11:1)

Indicates if input parity checking is enabled.

    0 - Input parity checking is disabled.
    1 - Input parity checking is enabled.

HW'LINE'SPEED(5).(12:4)

Indicates the current transfer rate of the ADCC.

    0 -   External - unused.
    1 -   External - unused.
    2 -   50 baud - unused.
    3 -   75 baud - unused.
    4 - 134.5 baud - unused.
    5 -   200 baud - unused.
    6 -   600 baud.
    7 -   2400 baud.
    8 -   9600 baud.
    9 -   4800 baud.
    10 - 1800 baud - unused.
    11 - 1200 baud.
    12 - 2400 baud.
    13 -   300 baud.
    14 -   150 baud.
    15 -   110 baud.

WORD 6

HW'OLD'INTERRUPT(6).(0:8)

Contains the old interrupt code. When an interrupt occurs the interrupt code in HW'INTERRUPT'CODE is moved here and the new interrupt is placed in HW'INTERRUPT'CODE. The first two bits show the CPVA number and the remaining bits are the interrupt code.

HW'INTERRUPT'CODE(6).(8:8)

Contains the last interrupt code. The first two bits are the CPVA number, and the last six bits are the interrupt code.

CPVA 0 -   Belated HIOP interrupt and channel program aborts.

CPVA 1 -   Speedsensing interrupts and ACK wait interrupts.

        0 - Unused.
        1 - 9600 baud or first part of 4800 baud.
        2 - First part of 2400 baud.
        3 - First part of 1200 baud.
        4 - First part of 600, 300, or 110 baud.
        5 - Second part of 600 or 300 baud.
        6 - Second part of 110 baud.
        7 - Last part of all but 9600 baud.
        8 - Non-speedsense detected by channel program.
        9 - Modem lines may have changed.
       10 - Special character received during ACK wait.

CPVA 2 -   Input Save state interrupts/Error conditions

        0 - Unused.
        1 - Input save buffer full.
        2 - Error condition during input save.
        3 - Special character during input save.
        4 - Error condition (overrun, parity, framing,
            break, modem change).

CPVA 3 -   Standard channel program interrupts.

        0 - Unused.
        1 - Unused.
        2 - End of set port protocol.
        3 - Unused.
        4 - Special character during write.
        5 - Unused.

---

        6 - Read complete.
        7 - Parity error detected.
        8 - Special character during read.
        9 - Framing error detected.
       10 - Overrun error detected.
       11 - Unused.
       12 - Unused.
       13 - Break detected.
       14 - Write complete.
       15 - Unused.
       16 - Edit special character found.
       17 - Unused.
       18 - Unused.
       19 - Unused.
       20 - Unused.
       21 - Unused.
       22 - Unused.
       23 - Unused.
       24 - Unused.
       25 - Unused.
       26 - Unused.
       27 - Unused.
       28 - Unused.
       29 - Modem line change detected.
       30 - Need to insert pad characters.
       31 - Special character during wait.
       32 - Trigger characters written.
       33 - An ENQ character has been written.
       34 - An ACK character has been written.
       35 - End of modem control channel program.

WORD 7

HW'READ'CNT

Current read byte count.

WORD 8

HW'RD'RIGHT'LEFT(8).(1:1)

Indicates what byte to start the transfer on.

    0 - Left.
    1 - Right.

HW'READ'BANK

Indicates the bank number for the current read.

WORD 9

HW'READ'ADDR

Contains the absolute start address of the read.

WORD 10

HW'WT'RIGHT'LEFT(10).(1:1)

Indicates what byte to start the write from.

    0 - Left.
    1 - Right.

HW'WRITE'BANK

Indicates the bank number for the current write.

WORD 11

HW'WRITE'ADDR

Contains the absolute start address of the write.

WORD 12

HW'WRITE'CNT

Current write byte count.

WORD 13

HW'DELAY'CHAR(13)(0:8)

Contains the last character (either output data or a trigger character) that requires delay syncs after it.

HW'DELAY'INT(13).(8:8)

Contains the interrupt code to generate after the delay syncs have been generated.

WORD 14

HW'FRAMING'ERROR(14).(0:8)

Contains a running count on the number of framing errors that occur. Used only for gathering information when dumps are read.

HW'SPEC'CHAR(14).(8:8)

Contains the last special character detected. Updated each time a new special character is detected.

WORD 15

HW'SENSE'TIMER

When the first interrupt during a speedsense occurs (the first part of the bit pattern has been received), a one-second timer is started to abort the sense if not completed during that time. This word is used to store the timer request index.

WORD 16

HW'NEXT'STATE(16).(0:4)

Indicates the next state of the RDCC.

HW'MASK'OFF(16).(8:8)

This word contains the bit pattern necessary to mask off all modem signals. This is used whenever a modem change is detected in order to cancel the channel service request until the next request to change the modem output signals.

WORD 17

HW'MODEM'REF(17).(0:8)

Contains an 8-bit modem reference mask. There is one bit for each input signal. If the signal is different from the reference and is a needed signal as specified by the control mask then an interrupt will occur.

    Bit 0 - Unused.
    Bit 1 - Unused.
    Bit 2 - 0 = reference (1 = mask).
    Bit 3 - Clear to send.
    Bit 4 - Data set ready.
    Bit 5 - Ring indicator.
    Bit 6 - Data carrier detect.
    Bit 7 - Secondary data carrier detect.

HW'MODEM'CTL(17).(8:8)

Contains an 8-bit modem control mask. There is one bit for each input sig-
nal. If the bit is set then signal will be used, otherwise it is ignored.
The mask is the same as in HW'MODEM'REF.

WORDS 18/25

HW'PRI'SCHRS

Contains an 128-bit mask of the primary special character set. There is
1-bit for each of the 128 ASCII characters. If set then the character is a
special character. The bit map has to be looked at as a 16-byte map where
the characters are numbered 0-7, left to right. For example if Control-A is
a special character the first word of the table would be X100000. If
backspace was set the first word would be X200.

WORDS 26/29

HW'SEC'SCHRS

Contains an 8-character buffer for a secondary special character set.

WORDS 30/33

HW'WRI'SCHRS

Contains an 8-character buffer for write special characters.

WORDS 34/37

HW'EDIT'SCHRS

Contains an 8-character buffer for write edit special character set.

WORD 38

HW'WRITE'BUFR

Contains a 2-character write buffer used to send trigger characters for
reads.

WORDS 39/46

HW'READ'BUFFER

Contains a 16-character read buffer used by the ADCC for idle reads.

WORD 47

Not currently used.

WORD 48

HW'SPEC'SPEED(48)

If the port is configured as subtype 4 or 5, this word contains the speed (in
CPS) at which we will initially set up the UART. This is the same as the
configured speed.

WORD 49

This word is currently used for performance measurements.

WORD 50

HW'ENABLE'XMIT(50).(0:8)

This field contains the bit pattern necessary to enable the transmitter buff-
er not full channel service request.

HW'DISABLE'XMIT(50).(8:8)

This field contains the bit pattern necessary to disable the transmitter
buffer not full channel service request.

WORD 51

HW'ECHO(51).(3:1)

This bit indicates the current state of echo during reads.

    0 - Echo is off.
    1 - Echo is on.

HW'MODEM'OUT1(51).(4:4)

This field contains the current setting of the modem output lines. When the
entire byte (0:8) is output to the ADCC, echo and modem lines are set. The
bits are used as follows:

    Bit 4 - Request to send.
    Bit 5 - Data terminal ready.
    Bit 6 - Speed select.
    Bit 7 - Secondary request to send.

HW'MODEM'OUT2(51).(12:4)

This field contains the identical information as HW'MODEM'OUT1. When the en-
tire byte (8:8) is output to the ADCC, the modem lines are set and echo is
turned off. This is used when transferring to the input save state from a
read.

WORD 52

HW'SAVE'READ(52).(0:8)

This field contains the character read whenever an error status is detected.
This is needed to distinguish a break from an ordinary framing error.

HW'STATUS'FE(52).(12:1)

This contains the framing error status bit read from the UART. It indicates
either a framing error or break.

HW'STATUS'OE(52).(13:1)

This contains the overrun error status bit read from the UART.

HW'STATUS'PE(52).(14:1)

This contains the parity error status bit read from the UART.

WORD 53

HW'UART(53).(0:8)

This field contains all the UART control information including character
width, parity, and stop bit information. It also has the bit set which
causes a master clear to be performed on the UART.

HW'SAVE'MODEM(53).(8:8)

This field contains the status of the modem lines being monitored. Only bits
3 through 7 of the byte are used and they have the same format as
HW'MODEM'REF above.

WORD 54

HW'INSAVE'OE(54).(0:1)

This bit is set true when an overrun error was detected during input save,
and cleared when the condition has been serviced.

HW'INSAVE'PE(54).(1:1)

This bit is used as above for the parity error condition.

HW'INSAVE'FE(54).(2:1)

This bit is used as above for the framing error condition.

HW'INSAVE'BREAK(54).(3:1)

This bit is used as above for the break condition.

HW'INSAVE'MODEM(54).(4:1)

This bit is used as above for a modem line change.

HW'INSAVE'NORCK(54).(5:1)

This bit is used as above for an ACK timeout.

HW'XON'WAIT(54).(14:1)

This bit indicates that an XOFF was received and the driver is waiting for an
XON to continue.

HW'ACK'WAIT(54).(15:1)

This bit indicates that an ENQ was sent and the driver is waiting for an ACK
to continue.

WORD 55

HW'ENQ'BLOCK(55).(0:8)

This field contains the block count of characters used in the ENQ/ACK hand-
shake. If the value is zero, the handshake is not used.

HW'ENQ'COUNT(55).(0:8)

If ENQ/ACK is enabled, this field contains the number of characters left to
write before another ENQ should be generated.

HW'CR'DELAY(55).(8:8)

If ENQ/ACK is disabled, this field contains the number of .1 seconds to delay
after CR characters.

WORD 56

HW'ENQ'CHAR(56).(0:?)

If ENQ/ACK is enabled, this field contains the Enquiry character.

HW'LF'DELAY(56).(0:8)

If ENQ/ACK is disabled, this field contains the number of .1 seconds to delay after LF characters.

HW'ACK'CHAR(56).(8:8)

If ENQ/ACK is enabled, this field contains the Acknowledge character.

HW'FF'DELAY(56).(8:8)

If ENQ/ACK is disabled, this field contains the number of .1 seconds to delay after FF characters.

WORD 57

HW'ENQ'TIMER(57)

When an Enquiry character has been written to the terminal, a ten-second timer is started. This word contains the timer request index for that timer.

WORD 58/73

HW'CHAR'MAP

These sixteen words contain a bit map which reflects the 256-bit map special character array in the hardware.

WORD 74/81

HW'INSAVE'BUF

These sixteen bytes are used as the input save buffer.

WORD 82/97

HW'CHAR'BUFFER

These thirty words are used as buffer space whenever the special character array in hardware must be changed.

Message Table Format

The message table is not used but contains the following:

"cr,lf,bell,bell","LDEV #1 NOT READY ","cf,lf"

Port Error Area Format

ATP/ATP37 Port Error Area Format

```
        0  1  2  3  4  5  6  7  8  9  10 11 12 13 14 15
     |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
   0| NUMBER OF CALLS TO LYNX'ERROR                    |
    |-------------------------------------------------|
   1|                                                 |
    |-------------------------------------------------|
  43| TERMINAL DATA SEGMENT HEADER - X43 WORDS         |
    |-------------------------------------------------|
  44|                                                 |
    |-------------------------------------------------|
  70| MONITOR DIT - X25 WORDS                          |
    |-------------------------------------------------|
  71|                                                 |
    |-------------------------------------------------|
 105| PROTOCOL AND DATA MANAGER FIXED DIT - X15 WORDS  |
    |-------------------------------------------------|
 106|                                                 |
    |-------------------------------------------------|
 221| PORT PROTOCOL DIT - X114 WORDS                   |
    |-------------------------------------------------|
 222|                                                 |
    |-------------------------------------------------|
 254| PROTOCOL AND DATA MANAGER VARIABLE DIT - X33 WORDS|
    |-------------------------------------------------|
 255|                                                 |
    |-------------------------------------------------|
 340| ATP/ATP37 HARDWARE DIT - X64 WORDS               |
    |-------------------------------------------------|
 341|                                                 |
    |-------------------------------------------------|
 352| TBUF TABLE - X12 WORDS                           |
    |-------------------------------------------------|
 353|                                                 |
    |-------------------------------------------------|
 372| ATP/ATP37 CONTROL PROGRAM - X20 WORDS            |
    |-------------------------------------------------|
 373|                                                 |
    |-------------------------------------------------|
 376| LPDT - X4 WORDS                                  |
    |-------------------------------------------------|
 377|                                                 |
    |-------------------------------------------------|
 406| DLT - X10 WORDS                                  |
    |-------------------------------------------------|
 407|                                                 |
    |-------------------------------------------------|
 412| DRT - X4 WORDS                                   |
    |-------------------------------------------------|
```

ATP/ATP37 Port Error Area Format (Cont.)

```
     |-------------------------------------------------|
 413|                                                 |
    |-------------------------------------------------|
 426| LDT - X7 WORDS AND LDTX (OR ZEROS IF NONE) - X5 WORDS|
    |-------------------------------------------------|
 427|                                                 |
    |-------------------------------------------------|
 453| IOQ PCB (OF ZEROS IF NONE) - X25 WORDS           |
    |-------------------------------------------------|
 454|                                                 |
    |-------------------------------------------------|
 500| LDT PCB (OR ZEROS IF NONE) - X25 WORDS           |
    |-------------------------------------------------|
 501|                                                 |
    |-------------------------------------------------|
 700| PCC MEMORY (OR ZEROS IF NONE) - X200 WORDS       |
    |-------------------------------------------------|
 701|                                                 |
    |-------------------------------------------------|
 720| ATP/ATP37 REGISTERS (OR ZEROS IF NONE) - X20 WORDS|
    |-------------------------------------------------|
 721|                                                 |
    |-------------------------------------------------|
1120| USERS STACK - X200 WORDS                         |
    |-------------------------------------------------|
1121|                                                 |
    |-------------------------------------------------|
1144| ILT - X16 WORDS AND ILTX - X6 WORDS              |
    |-------------------------------------------------|
1145|                                                 |
    |-------------------------------------------------|
1150| VFC SIR'S - X4 WORDS                             |
    |-------------------------------------------------|
1151|                                                 |
    |-------------------------------------------------|
1152| VFC INFORMATION BLOCK - X2 WORDS                 |
    |-------------------------------------------------|
    |                                                 |
    |-------------------------------------------------|
    | VFC ENTRY (OR NOTHING IF NONE) - X20 WORDS       |
    |-------------------------------------------------|
    |                                                 |
    |-------------------------------------------------|
    | VFC BUFFER (OR NOTHING IF NONE) - X105 WORDS     |
    |-------------------------------------------------|
    |                                                 |
    |-------------------------------------------------|
    | IOQ AND TBUF INFORMATION BLOCK - X3 WORDS        |
    |-------------------------------------------------|
    |                                                 |
    |-------------------------------------------------|
    | IOQ'S - X14 WORDS EACH AND TBUF'S/SBUF'S - X105/X200 WORDS|
    |-------------------------------------------------|
```

The VFC information block contains the following:

Word 0.(0:1) - 0: No VFC entry or data dumped.
              1: A VFC entry was dumped.
      .(8:4) - Number of initialization buffers dumped.
      .(12:4) - Number of data buffers dumped.

Word 2   - TDS relative pointer to IOQ/TBUF information block.

The IOQ/TBUF information block contains the following:

Word 0   - Number of IOQs dumped.

Word 1   - TDS relative pointer first TBUF dumped.

Word 2.(0:2) - 0: No TBUFs or SBUFs were dumped.
              1: TBUFs were dumped.
              2: SBUFs were dumped.
      .(2:14) - Number of TBUFs or SBUFs dumped.

ADCC Port Error Area Format

```
        0  1  2  3  4  5  6  7  8  9  10 11 12 13 14 15
       |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
     0| NUMBER OF CALLS TO LYNX'ERROR                    |
      |-------------------------------------------------|
     1|                                                 |
      |-------------------------------------------------|
    43| TERMINAL DATA SEGMENT HEADER - X43 WORDS         |
      |-------------------------------------------------|
    44|                                                 |
      |-------------------------------------------------|
    70| MONITOR DIT - X25 WORDS                          |
      |-------------------------------------------------|
    71|                                                 |
      |-------------------------------------------------|
   105| PROTOCOL AND DATA MANAGER FIXED DIT - X15 WORDS  |
      |-------------------------------------------------|
   106|                                                 |
      |-------------------------------------------------|
   221| PORT PROTOCOL DIT - X114 WORDS                   |
      |-------------------------------------------------|
   222|                                                 |
      |-------------------------------------------------|
   254| PROTOCOL AND DATA MANAGER VARIABLE DIT - X33 WORDS|
      |-------------------------------------------------|
   255|                                                 |
      |-------------------------------------------------|
   416| ADCC HARDWARE DIT - X142 WORDS                   |
      |-------------------------------------------------|
   417|                                                 |
      |-------------------------------------------------|
   430| TBUF TABLE - X12 WORDS                           |
      |-------------------------------------------------|
   431|                                                 |
      |-------------------------------------------------|
   746| CHANNEL PROGRAM - X316 WORDS                     |
      |-------------------------------------------------|
   747|                                                 |
      |-------------------------------------------------|
   752| LPDT - X4 WORDS                                  |
      |-------------------------------------------------|
   753|                                                 |
      |-------------------------------------------------|
   762| DLT - X10 WORDS                                  |
      |-------------------------------------------------|
   763|                                                 |
      |-------------------------------------------------|
   766| DRT - X4 words                                   |
      |-------------------------------------------------|
```

ADCC Port Error Area Format (Cont.)

```
   767|                                                     |
      |-----------------------------------------------------|
  1002| LDT - X7 WORDS AND LDTX (OR ZEROS IF NONE) - X5 WORDS|
      |-----------------------------------------------------|
  1003|                                                     |
      |-----------------------------------------------------|
  1027| IOQ PCB (OF ZEROS IF NONE) - X25 WORDS               |
      |-----------------------------------------------------|
  1030|                                                     |
      |-----------------------------------------------------|
  1054| LDT PCB (OR ZEROS IF NONE) - X25 WORDS               |
      |-----------------------------------------------------|
  1055|                                                     |
      |-----------------------------------------------------|
  1254| USERS STACK - X200 WORDS                             |
      |-----------------------------------------------------|
  1255|                                                     |
      |-----------------------------------------------------|
  1300| ILT - X16 WORDS AND ILTX - X6 WORDS                  |
      |-----------------------------------------------------|
  1301|                                                     |
      |-----------------------------------------------------|
  1304| VFC SIR'S - X4 WORDS                                 |
      |-----------------------------------------------------|
  1305|                                                     |
      |-----------------------------------------------------|
  1306| VFC INFORMATION BLOCK - X2 WORDS                     |
      |-----------------------------------------------------|
       | VFC ENTRY (OR NOTHING IF NONE) - X20 WORDS          |
      |-----------------------------------------------------|
       | VFC BUFFER (OR NOTHING IF NONE) - X105 WORDS        |
      |-----------------------------------------------------|
       | IOQ AND TBUF INFORMATION BLOCK - X3 WORDS           |
      |-----------------------------------------------------|
       | IOQ'S - X14 WORDS EACH AND TBUF'S/SBUF'S - X105/X200 WORDS |
      |-----------------------------------------------------|
```

The VFC information block contains the following:

Word 0.(0:1) - 0: No VFC entry or data dumped.
              1: A VFC entry was dumped.
     .(8:4)   - Number of initialization buffers dumped.
     .(12:4) - Number of data buffers dumped.
Word 2   - TDS relative pointer to IOQ/TBUF information block.

The IOQ/TBUF information block contains the following:

Word 0   - Number of IOQs dumped.
Word 1   - TDS relative pointer first TBUF dumped.
Word 2.(0:2) - 0: No TBUFs or SBUFs were dumped.
              1: TBUFs were dumped.
              2: SBUFs were dumped.
     .(2:14) - Number of TBUFs or SBUFs dumped.

TBUF Table Format

```
        0  1  2  3  4  5  6  7  8  9  10 11 12 13 14 15
       |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
     0| NUMBER OF TBUF'S IN TDS                          |
      |-------------------------------------------------|
     1| TBUF'S SAVED FOR READS   | SIZE OF TBUF IN WORDS |
      |-------------------------------------------------|
     2| TDS RELATIVE POINTER TO HEAD OF FREE TBUF LIST   |
      |-------------------------------------------------|
     3| TDS RELATIVE POINTER TO TAIL OF FREE TBUF LIST   |
      |-------------------------------------------------|
     4| MAXIMUM NUMBER OF TBUF'S EVER IN USE             |
      |-------------------------------------------------|
     5| CURRENT NUMBER OF TBUF'S IN USE                  |
      |-------------------------------------------------|
     6|                                                 |
     7| TOTAL NUMBER OF TBUF REQUESTS                    |
      |-------------------------------------------------|
    10| NUMBER OF TBUF REQUESTS DENIED                   |
      |-------------------------------------------------|
    11| UNUSED                                           |
      |-------------------------------------------------|
```

WORD 0

TBUF'NUM'WRD

Contains the number of TBUFs in the data segment.

WORD 1

TBUF'READ'SAVE(1).(0:8)

Contains the number of TBUFs saved for reads.

TBUF'BUFSIZE(1).(8:8)

Indicates the size in words of each TBUF.

WORD 2

TBUF'LISTHEAD'P

Contains a TDS relative pointer to the head TBUF in the TBUF free list.

WORD 3

TBUF'LISTTRAIL'P

Contains a TDS relative pointer to the tail TBUF in the TBUF free list.

WORD 4

TBUF'MAXUSED

Indicates the maximum number of TBUFS in use at any time.

WORD 5

TBUF'INUSE'WRD

Indicates the number of TBUFs currently in use.

WORDS 6/7

TOTALREQUESTS

Indicates the total number of TBUF requests.

WORD 8

TBUF'DENIED'WRD

Indicates the number of TBUF requests that were denied because
there were no free TBUFS.

WORD 9

Not currently used.

TBUF Format

```
    |----------------------------------------------------------|
  0| TDS TABLE RELATIVE POINTER TO NEXT TBUF - 0 IF NO LINK    |
    |----------------------------------------------------------|
  1|                                                          |
    
103| USER DATA                                                |
    |----------------------------------------------------------|
104| 0 - NEVER USED                                           |
    |----------------------------------------------------------|
```

Terminal Monitor DIT Format

```
     0   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15
    |---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
    |TE |   |ACT|   |RE |BRO|   |TBU| TRUE HARDWARE UNIT NUMBER      |
  0 |RM |UP |IVE|REQ|SET|KEN|   |FAV| OF PORT (0 - 127)             |
    |---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
  1| SYS DB RELATIVE POINTER TO NEXT DIT WAITING FOR SYSIO         |
    |--------------------------------------------------------------|
  2| IOQ TABLE RELATIVE POINTER TO HEAD IOQ                        |
    |--------------------------------------------------------------|
  3| LOGICAL DEVICE NUMBER                                         |
    |--------------------------------------------------------------|
  4| SYS DB RELATIVE POINTER TO DRIVER LINKAGE TABLE              |
    |--------------------------------------------------------------|
  5| SYS DB RELATIVE POINTER TO INTERRUPT LINKAGE TABLE          |
    |---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
    |HAN|   |CFA|   |   |   |   |RE |   |LOG|   |   |   |          |
  6|GUP|   |IL |   |XON|   |   |AD |   |ON |   |   |ACK|          |
    |---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
  7| INTERRUPT MANAGER COMMUNICATION WORD                         |
    |---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
    |PRE|PRE|BIN|SPD|  WAIT   |FLU| LOGON |NO |CON|PF | PREEMPT    |
 10 |EMP|SPA|ARY|SNS| REASON  |SH | TYPE  |CAT|MOD|REC| LEVEL      |
    |---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
 11| TERMINAL DST RELATIVE POINTER TO PROTOCOL & DATA MANAGER DIT |
    |--------------------------------------------------------------|
 12|                          |CONFIGURED UNIT NUMBER (0 - 95)    |
    |---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
 13|P/F| PCC DATE CODE |P/F| MCC DATE CODE |P/F| MSC DATE CODE |JPT|
    |---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
 14| DEVICE TYPE SUPPORTED | DUMMY  DRIVER   |CONTROL|   |NO |BRK|
    | BY THIS DRIVER        | VERSION NUMBER  |TYPE   |SS |FIX|MOD|
    |-----------------------|-----------------|-------|---|---|---|
 15| RESERVED FOR SYSTEM LOGGING                                   |
    |--------------------------------------------------------------|
 16| RESERVED FOR SYSTEM LOGGING                                   |
    |--------------------------------------------------------------|
 17| ERROR CODE                                                   |
    |---|----------------------------------------------------------|
 20|TIM| NEXT READ TIME OUT VALUE - 10THS/SEC                     |
    |---|----------------------------------------------------------|
 21| SAVED IOQ PARAMETER                                          |
    |--------------------------------------------------------------|
 22| LOGON TIME OUT INDEX                                          |
    |--------------------------------------------------------------|
 23| TEMP STORAGE FOR MONITOR                                      |
    |--------------------------------------------------------------|
 24| LAST TIMED READ VALUE                                         |
    |--------------------------------------------------------------|
```

WORD 0

DL'TERM(0).(0:1)

Indicates if the device is a terminal. Always a one.

    0 - Device is not a terminal.
    1 - Device is a terminal.

DL'UP(0).(1:1)

Indicates if the device is "up".

    0 - Device is not up.
    1 - Device is up and has been speedsensed or FOPEN'ed.

DL'ACTIVE(0).(2:1)

Indicates the monitor is active.

0 - The monitor is not active. 1 - The monitor is active and processing a
function.

DL'REQUEST(0).(3:1)

Indicates if the monitor was awakened while active.

    0 - There is no pending request.
    1 - The monitor was awakened while active and has a pending
        request.

DL'RESET(0).(4:1)

Indicates if the monitor should reinitialize the port.

    0 - Don't reinitialize the port.
    1 - Reinitialize the port. This is equivalent to doing
        an ABORTJOB or device close against the port.

DL'BROKEN(0).(5:1)

Indicates if the port is broken.

    0 - The port is not broken.
    1 - The driver detected an error and marked the port
        broken. The error code will be in DL'ERROR'CODE.

DL'TBUFAVAIL(0).(7:1)

Indicates if a TBUF is now available for the device.

    0 - No meaning.
    1 - A TBUF(s) is available and the write can resume.

DL'TRUEUNIT(0).(8:8)

Contains the true unit number of the device. Is only used for devices con-
nected to an ATP/ATP37 controller. Unit numbers will range from 0 to 127.

WORD 1

DL'NEXT

Contains a SYS DB relative pointer to the next DIT waiting for SYSIO/TERMIO.

WORD 2

DL'IOQP

Contains a IOQ table relative pointer to the current IOQ.

WORD 3

DL'LDEV

Contains the configured logical device number of the device.

WORD 4

DL'DLTP

Contains a SYS DB relative pointer to the driver linkage table.

WORD 5

DL'ILTP

Contains a SYS DB relative pointer to the interrupt linkage table.

WORD 6

DL'TICK

The TICK communication word. If a timer is running and expires the bit cor-
responding to the type of timer is set in DL'TICK. The monitor is then
awakened to process the timeout.

DL'HANGUP'TO(6).(0:1)

Timer used by the Initiation Manager when disconnecting a modem.

DL'CFAIL'TO(6).(2:1)

Timer used by the Initiation Manager and Interrupt Manager when trying to connect a modem, waiting for DSR and DCD to come on or when there is a carrier fail.

DL'XON'TO(6).(4:1)

Timer used by the Interrupt Manager when waiting for an XON.

DL'READ'TO(6).(8:1)

Timer used by the Initiation Manager and Interrupt Manager for reads.

DL'LOGON'TO(6).(11:1)

Timer used by the monitor for a logon timeout.

DL'ACK'TO(6).(14:1)

Timer used by the RDCC physical driver for a 10 ENQ/ACK timeout.

WORD 7

DL'INT'MAN

The Interrupt Manager communication word. When the Interrupt Manager needs to awaken the monitor it will place an interrupt code in this word and then awaken the monitor via AWAKETERMINAL. This word contains 4 4-bit fields so that the Interrupt Manager may awaken the monitor for more that one reason. The fields are processed by the monitor left to right. Interrupt codes are as follow:

1 - Disconnect interrupt. Data set ready has dropped or carrier fail has occurred more that 50 times during the read. The monitor will initiate a disconnect sequence of the modem.
2 - Partial hardware setup. The hardware has been partially set up and the monitor will call the Initiation Manager to continue/finish setting up the hardware.
3 - Partial read interrupt. The read has been completed or there were no TBUFs available to complete the read. The monitor will call the Initiation Manager to complete the read.
4 - Partial write interrupt. The write was a critical write or there was an "ACK" time out or tanking needs to be resumed on the write. The monitor will call the Initiation Manager to continue/complete the write.
5 - Speedsense interrupt. The device has successfully been speedsensed. The monitor will awaken "DEVREC" for device recognition and then call the Initiation Manager to initialize the port.

6 - Subsystem break interrupt. A subsystem break has been detected. The monitor will call "BREAKSS" to see if MPE will accept the subsystem break. The Initiation Manager is then called. If reading the read data is returned. If writing the write data is flushed. The flush bit is then set in all linked IOQs. If the subsystem break is not accepted the current operation is resumed.
7 - Operation done. The current write is complete and the monitor needed to be notified.
8 - Break interrupt. A break interrupt has been detected. The monitor will call "BREAKJOB" to see if MPE will accept the break. If accepted the Initiation Manager is called. If reading the read data is returned. If writing all write data is flushed. The IOQs are then marked broken. If the break is not accepted the current operation is resumed.
9 - Reset done. The reset of the port is complete. The monitor will either continue with the "open" if in the middle of the "open" or start of a speedsense.

WORD 8

DL'PREEMPT(8).(0:1)

Indicates if there is a preemptive request.

0 - No preemptive request.
1 - Set by ATTACHIO if there is a preemptive IOQ linked in the DIT.

DL'PRESPACE(8).(1:1)

Indicates if the driver is prespacing writes.

0 - Prespacing is not enabled.
1 - The last IOQ specified prespacing.

DL'BINARY(8).(2:1)

Indicates if in binary mode.

0 - Not in binary mode.
1 - Binary mode.

DL'SPD'SNS(8).(3:1)

Indicates if the device has been speedsensed.

0 - The device has not been speedsensed.
1 - The device has been speedsensed.

DL'WAIT'RSN(8).(4:3)

This indicates that the monitor is waiting to be awakened by the Interrupt Manager. Wait reasons are as follow:

0 - Not waiting.
1 - Disconnect complete.
2 - Hardware set up complete.
3 - Preempt complete.
4 - Partial IOQ complete.
5 - IOQ complete.
6 - Hard reset complete.
7 - Unused.

DL'FLUSH(8).(7:1)

Indicates if break was accepted by "BREAKJOB".

1 - Break was accepted and all IOQs should be flushed until a clear flush and write request is processed.

DL'LOGON'TYP(8).(8:2)

Indicates the logon type.

0 - Data accepting device.
1 - Session.
2 - Job.

DL'DONT'CAT(8).(10:1)

DL'CONSOLE(8).(11:1)

Indicates if in console mode.

0 - Not in console mode.
1 - Device is in console mode.

DL'PF'RECV(8).(12:1)

Indicates if a power fail occurred and the driver should go through its power fail processing.

0 - No power fail occurred.
1 - A power fail has occurred and the driver will process the power fail.

DL'PRP'LEVEL(8).(13:3)

Contains the preempt level of the current write.

WORD 9

DL'PD'DITP

Contains a TDS relative pointer to the Protocol and Data Manager DIT.

WORD 10

DL'UNIT(10).(8:8)

Contains the configured logical unit number of the device. It is only used for devices connected to the ATP/ATP37 controller. Unit numbers will range from 0 to 95.

WORD 11

DL'DATE'CODE

Contains information that indicates the level of hardware. For an ATP controller this will contain the date codes of the 6801's and the results from selftest. For the RDCC this is unused.

WORD 12

DL'DEVTYPE(12).(0:6)

Indicates the driver type. For terminals the driver type will be 16.

DL'VERSION(12).(6:5)

Indicates the current version of the driver. For ATP the driver is HIOTERM1/HIOASLP0 and for the RDCC the driver is HIOTERM2/HIOASLP2.

DL'CONTROLLER(12).(11:2)

Indicates the type of controller used by this LDEV.

1 - ATP.
2 - RDCC.

DL'SSTO(12).(13:1)

Indicates how to process the logon timer that is running.

0 - If the logon timer expires the logon did not occur and the port should be disconnected.
1 - If the logon timer expires the speedsense should be aborted and restarted.

DL'UNFIXABLE(12).(14:1)

Indicates if the port is broken and unfixable.

    0 - Port is not unfixable, but may be broken.
    1 - Port is unfixable. A warmstart of the system will reset the port.

DL'BRK'MODE(12).(15:1)

Indicates if the device is in break mode.

    0 - The device is not in break mode. Cleared with an IOQ
        function code of 31.
    1 - The device is in break mode. Set with an IOQ function
        code of 31.

WORD 13

DL'LOG

Used for system logging. When a port failure occurs LYNX'ERROR will save in here the status register from the stack marker.

WORD 14

DL'LOG1

Used for system logging. When a port failure occurs LYNX'ERROR will save in here the P register from the stack marker.

WORD 15

DL'ERROR'CODE

Contains the error code when a port failure occurs. The code is placed here by LYNX'ERROR when the failure occurs. It is a 4-digit (decimal) code where the first two digits are the module number in which the error occurred and the second two are a unique error code.

WORD 16

DL'TIME'FLAG(16).(0:1)

Indicates if the next read will be timed.

    0 - The next read is not timed. Cleared with an IOQ function
        code of 16.
    1 - The next read will be timed. Set with an IOQ function
        code of 17.

DL'READ'TVAL(16).(1:15)

Contains the time out value for the next read that is to be timed. Set with an IOQ function code of 5

DL'TIME

A reference to both read time fields.

WORD 17

DL'QPARM

Saved IOQ parameter.

WORD 18

DL'LOGN'TRLX

Contains the logon timer index.

WORD 20

DL'TEMP

Temporary storage used by monitor

WORD 19

DL'READ'TIME

For reads that are timed, DL'TIME'FLAG = 1, this will contain the read time for the last read.

Line Printer Monitor DIT Format

```
     0  1  2  3  4  5  6  7  8  9  10 11 12 13 14 15
    |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
    |TE |   |ACT|   |RE |BRO|   |TBU| TRUE HARDWARE UNIT NUMBER |
   0|RM |UP |IVE|REQ|SET|KEN|   |FAV| OF PORT (0 - 127)         |
    |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
   1| SYS DB RELATIVE POINTER TO NEXT DIT WAITING FOR SYSIO     |
    |----------------------------------------------------------|
   2| IOQ TABLE RELATIVE POINTER TO HEAD IOQ                    |
    |----------------------------------------------------------|
   3| LOGICAL DEVICE NUMBER                                     |
    |----------------------------------------------------------|
   4| SYS DB RELATIVE POINTER TO DRIVER LINKAGE TABLE           |
    |----------------------------------------------------------|
   5| SYS DB RELATIVE POINTER TO INTERRUPT LINKAGE TABLE        |
    |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
    |HAN|   |CFA|   |   |   |RE |   |   |LOG|   |   |   |        |
   6|GUP|   |IL |   |XON|   |RD |   |   |ON |   |   |RCK|        |
    |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
   7| INTERRUPT MANAGER COMMUNICATION WORD                      |
    |--|-----------|--|--|--|-----------------|--|--|-----------|
  10|RES|          |PRE|NO |   |                |PF |           |
    |SET|          |SPA|STP|WAT|                |REC|           |
    |--|-----------|--|--|--|-----------------|--|--|-----------|
  11| TERMINAL DST RELATIVE POINTER TO PROTOCOL & DATA MANAGER DIT|
    |----------------------------------------------------------|
  12|                        |CONFIGURED UNIT NUMBER (0 - 95)   |
    |--|------------|--|------------|--|------------|--|--------|
  13|P/F| PCC DATE CODE |P/F| MCC DATE CODE |P/F| MSC DATE CODE |JPT|
    |--|------------|--|------------|--|------------|--|--------|
  14| DEVICE TYPE SUPPORTED  | DUMMY  DRIVER  |CONTROL| |NO |   |
    | BY THIS DRIVER         | VERSION NUMBER |TYPE   | |FIX|   |
    |--|------------|--|------------|--|------------|--|--------|
  15| RESERVED FOR SYSTEM LOGGING                               |
    |----------------------------------------------------------|
  16| RESERVED FOR SYSTEM LOGGING                               |
    |----------------------------------------------------------|
  17| ERROR CODE                                                |
    |----------------------------------------------------------|
  20| TEMP STORAGE FOR MONITOR                                  |
    |----------------------------------------------------------|
```

WORD 0

DL'TERM(0).(0:1)

Indicates if the device is a terminal. Always a one.

    0 - Device is not a terminal.
    1 - Device is a terminal.

DL'UP(0).(1:1)

Indicates if the device is "up".

    0 - Device is not up.
    1 - Device is up and has been FOPEN'ed.

DL'ACTIVE(0).(2:1)

Indicates the monitor is active.

    0 - The monitor is not active.
    1 - The monitor is active and processing a function.

DL'REQUEST(0).(3:1)

Indicates if the monitor was awakened while active.

    0 - There is no pending request.
    1 - The monitor was awakened while active and has a pending request.

DL'RESET(0).(4:1)

Indicates if the monitor should reinitialize the port.

    0 - Don't reinitialize the port.
    1 - Reinitialize the port. This is equivalent to doing
        an ABORTJOB or device close against the port.

DL'BROKEN(0).(5:1)

Indicates if the port is broken.

    0 - The port is not broken.
    1 - The driver detected an error and marked the port
        broken. The error code will be in DL'ERROR'CODE.

DL'TBUFAVAIL(0).(7:1)

Indicates if a TBUF is now available for the device.

DL'TRUEUNIT(0).(8:8)

Contains the true unit number of the device. Is only used for devices connected to an ATP controller. Unit numbers will range from 0 to 127.

WORD 1

DL'NEXT

WORD 2

DL'IOQP

Contains an IOQ table relative pointer to the current IOQ.

WORD 3

DL'LDEV

Contains the configured logical device number of the device.

WORD 4

DL'DLTP

Contains a SYS DB relative pointer to the driver linkage table.

WORD 5

DL'ILTP

Contains a SYS DB relative pointer to the interrupt linkage table.

WORD 6

DL'TICK

The TICK communication word. If a timer is running and expires the bit corresponding to the type of timer is set in DL'TICK. The monitor is then awakened to process the timeout.

DL'HANGUP'TO(6).(0:1)

Timer used by the Initiation Manager when disconnecting a modem.

DL'CFAIL'TO(6).(2:1)

Timer used by the Initiation Manager and Interrupt Manager when trying to connect a modem, waiting for DSR and DCD to come on or when there is a carrier fail.

DL'XON'TO(6).(4:1)

Timer used by the Interrupt Manager when waiting for an XON.

DL'READ'TO(6).(8:1)

Timer used by the Initiation Manager and Interrupt Manager for reads.

DL'LOGON'TO(6).(11:1)

Timer used by the monitor for a logon timeout.

DL'ACK'TO(6).(14:1)

Timer used by the ADCC physical driver for a 10 ENQ/ACK timeout.

WORD 7

DL'INT'MAN

The Interrupt Manager communication word. When the Interrupt Manager needs to awaken the monitor it will place an interrupt code in this word and then awaken the monitor via AWAKETERMINAL. This word contains 4 4-bit fields so that the Interrupt Manager may awaken the monitor for more than one reason. The fields are processed by the monitor left to right. Interrupt codes are as follows:

    1 - Disconnect interrupt. Data set ready has dropped or carrier
        fail has occurred more that 50 times during the read. The
        monitor will initiate a disconnect sequence of the modem.

    2 - Partial hardware setup. The hardware has been partially set
        up and the monitor will call the Initiation Manager to
        continue/finish setting up the hardware.

    4 - Partial write interrupt. The write was a critical write
        or there was an "ACK" time out or tanking needs to be
        resumed on the write. The monitor will call the
        Initiation Manager to continue/complete the write.

    9 - Reset done. The reset of the port is complete. The
        monitor will either continue with the "open" if in the
        middle of the "open" or start a speedsense.

WORD 8

DL'RESETTING(8).(0:1)

Set if the port should be reset.

DL'PRESPACE(8).(4:1)

Set if the last request used prespacing.

DL'NO'STP'OV(8).(5:1)

Set if the driver does not perform auto perforation skip.

DL'WAIT(8).(6:1)

Set if the monitor is waiting for a request to complete.

DL'PF'REC(8).(12:1)

Set if the monitor should execute its power fail recovery code.

WORD 9

DL'PD'DITP

Contains a TDS relative pointer to the Protocol and Data Manager DIT.

WORD 10

DL'UNIT(10).(8:8)

Contains the configured logical unit number of the device. It is only used for devices connected to the ATP controller. Unit numbers will range from 0 to 95.

WORD 11

DL'STATE'CODE

Contains information that indicates the level of hardware. For an ATP controller this will contain the data codes of the 6801's and the results from selftest. For the ADCC this is unused.

WORD 12

DL'DEVTYPE(12).(0:6)

Indicates the driver type. For printers the driver type will be 32.

DL'VERSION(12).(6:5)

Indicates the current version of the driver. For ATP the driver is HIOTERM1 and for ADCC the driver is HIOTERM2.

DL'CONTROLLER(12).(11:2)

Indicates the type of controller used by this LDEV.

    1 - ATP.
    2 - ADCC.

DL'UNFIXABLE(12).(14:1)

Indicates if the port is broken and unfixable.

    0 - Port is not unfixable, but may be broken.
    1 - Port is unfixable. A warmstart of the system will reset the port.

WORD 13

DL'LOG

Used for system logging. When a port failure occurs LYNX'ERROR will save in here the status register from the stack marker.

WORD 14

DL'LOG1

Used for system logging. When a port failure occurs LYNX'ERROR will save in here the P register from the stack marker.

WORD 15

DL'ERROR'CODE

Contains the error code when a port failure occurs. The code is placed here by LYNX'ERROR when the failure occurs. It is a 4-digit (decimal) code where the first two digits are the module number in which the error occurred and the second two are a unique error code.

WORD 16

DL'TEMP

Temporary storage used by monitor.

ILT/ILTX Format

RTP/ATP37 ILT Format

```
    0  1  2  3  4  5  6  7  8  9  10 11 12 13 14 15
   |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
 0 |                                                |
 1 | CHANNEL PROGRAM VARIABLE AREA - CPVA            |
 2 | (NOT USED - SET TO 0)                           |
 3 |                                                 |
   |-------------------------------------------------|
 4 | DMA ABORT ADDRESS                               |
 5 | (NOT USED - SET TO 0)                           |
   |-------------------------------------------------|
 6 | (NOT USED - SET TO 0)                           |
   |--|------------------|----|----------|-----------|
 7 |M | CHANNEL QUEUE #   | IP | CHANNEL  | DEVICE    |
   |--|------------------|----|----------|-----------|
10 | CHANNEL PROGRAM POINTER - ILTX                  |
   |-------------------------------------------------|
11 | STATUS RETURN AREA POINTER - (NOT USED - SET TO 0)|
   |-------------------------------------------------|
12 | UNIT EXTRACT INSTRUCTION - (NOT USED - SET TO 0) |
   |-------------------------------------------------|
13 | CURRENT DIT POINTER - (NOT USED - SET TO 0)      |
   |---------------------------|---------------------|
14 | SIOP SIZE                 | CQUEN               |
   |---------------------------|---------------------|
15 | CONTROLLER FLAGS - (SET TO 0) | HIGHEST UNIT #   |
   |-------------------------------------------------|
16 | SYSDB RELATIVE DIT POINTER FOR UNIT 0 ON SIB    |
   |                                                 |
   |                                                 |
   | SYSDB RELATIVE DIT POINTER FOR UNIT N ON SIB    |
   |-------------------------------------------------|
```

WORD X7

ILT'DRT.(7:9)

Contains DRT number for controller on channel.

WORD X10

ILT'ISIOP

Contains a SYSDB relative pointer to the channel program area also known as the ILTX.

WORD X14

ILT'SIOP'SIZE.(0:8)

Contains size, in words, of channel program area (ILTX).

WORD X15

ILT'HUNIT.(9:7)

Contains the highest configured unit number on the channel.

WORD X16

Starting at word X16 there is a SYSDB relative pointer to the monitor DIT for each unit configured on the channel.

RTP/ATP37 ILTX Format

```
   |-------------------------------------------------|
 0 | TERMINAL DATA SEGMENT DST NUMBER FOR 1ST DST    |
   |-------------------------------------------------|
 1 | TERMINAL DATA SEGMENT BANK NUMBER FOR 1ST DST   |
   |-------------------------------------------------|
 2 | TERMINAL DATA SEGMENT DST OFFSET FOR 1ST DST    |
   |---|---------------------------------------------|
 3 |ID | (NOT USED - SET TO 0)                       |
   |---|---------------------------------------------|
 4 | CONTROLLER ID                                   |
   |-------------------------------------------------|
 5 | INTERRUPT PROCESSOR PLABEL                      |
   |-------------------------------------------------|
 6 | TERMINAL DATA SEGMENT DST NUMBER FOR 2ND DST    |
   |-------------------------------------------------|
 7 | TERMINAL DATA SEGMENT BANK NUMBER FOR 2ND DST   |
   |-------------------------------------------------|
10 | TERMINAL DATA SEGMENT DST OFFSET FOR 2ND DST    |
   |---|---------------------------------------------|
11 |ID | (NOT USED - SET TO 0)                       |
   |---|---------------------------------------------|
12 | CONTROLLER ID                                   |
   |-------------------------------------------------|
13 | INTERRUPT PROCESSOR PLABEL                      |
   |-------------------------------------------------|
```

RDCC ILT Format

```
    0  1  2  3  4  5  6  7  8  9  10 11 12 13 14 15
   |--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
 0 |                                                |
 1 | CHANNEL PROGRAM VARIABLE AREA - CPVA            |
 2 |                                                 |
 3 |                                                 |
   |-------------------------------------------------|
 4 | DMA ABORT ADDRESS                               |
 5 | (NOT USED - SET TO 0)                           |
   |-------------------------------------------------|
 6 | (NOT USED - SET TO 0)                           |
   |--|------------------|----|----------|-----------|
 7 |M | CHANNEL QUEUE #   | IP | CHANNEL  | DEVICE    |
   |--|------------------|----|----------|-----------|
10 | CHANNEL PROGRAM POINTER - ILTX                  |
   |-------------------------------------------------|
11 | STATUS RETURN AREA POINTER - (NOT USED - SET TO 0)|
   |-------------------------------------------------|
12 | UNIT EXTRACT INSTRUCTION - (NOT USED - SET TO 0) |
   |-------------------------------------------------|
13 | CURRENT DIT POINTER - (NOT USED - SET TO 0)      |
   |---------------------------|---------------------|
14 | SIOP SIZE                 | CQUEN               |
   |---------------------------|---------------------|
15 | CONTROLLER FLAGS - (SET TO 0) | HIGHEST UNIT #   |
   |-------------------------------------------------|
16 | SYSDB RELATIVE DIT POINTER                      |
   |-------------------------------------------------|
```

RDCC ILTX Format

```
   |-------------------------------------------------|
 0 | TERMINAL DATA SEGMENT DST NUMBER                |
   |-------------------------------------------------|
 1 | TERMINAL DATA SEGMENT BANK NUMBER               |
   |-------------------------------------------------|
 2 | TERMINAL DATA SEGMENT DST OFFSET                |
   |---|---------------------------------------------|
 3 |ID | (NOT USED - SET TO 0)                       |
   |---|---------------------------------------------|
 4 | CONTROLLER ID                                   |
   |-------------------------------------------------|
 5 | INTERRUPT PROCESSOR PLABEL                      |
   |                                                 |
 6 |                                                 |
   |                                                 |
315| RDCC CHANNEL PROGRAM                            |
   |-------------------------------------------------|
```

READER COMMENT SHEET

MPE V Tables Manual for MPE V, Version G.23.00

32033-90147   October 1991

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   Yes [] No []   (If no, explain under Comments, below.)
understand?

Is the format of this manual convenient   Yes [] No []   (If no, explain or suggest improvements
in size, arrangement and readability?   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   _____

_____

_____

BUSINESS REPLY MAIL

FIRST CLASS   PERMIT NO. 1070   CUPERTINO, CALIFORNIA

POSTAGE WILL BE PAID BY ADDRESSEE

Documentation Manager
Hewlett-Packard Company
Software Technology Division
8010 Foothills Blvd
Roseville, California 95678
ATTN: Documentation Manager: R5YF

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 _____

_____

_____

**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
  Hewlett-Packard Company
  Software Technology Division
  8010 Foothills Blvd
  Roseville, California 95678
  ATTN: Documentation Manager: R5YF

READER COMMENT SHEET

MPE V Tables Manual for MPE V, Version G.23.00

32033-90147   October 1991

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 _____

_____

_____

BUSINESS REPLY MAIL

FIRST CLASS   PERMIT NO. 1070   CUPERTINO, CALIFORNIA

POSTAGE WILL BE PAID BY ADDRESSEE

Documentation Manager
Hewlett-Packard Company
Software Technology Division
8010 Foothills Blvd
Roseville, California 95678
ATTN: Documentation Manager: R5YF

HEWLETT
PACKARD