205 pgs

1983 9943

Burroughs Corporation
COMPUTER SYSTEMS GROUP
PASADENA PLANT

MCP TABLES

COMPANY CONFIDENTIAL

PRODUCT SPECIFICATION

REVISIONS

REV LTR	REVISION ISSUE DATE	PAGES REVISED, ADDED, DELETED OR CHANGE OF CLASSIFICATION	PREPARED BY	APPROVED BY
A	5/31/83	REVIEW	G. Boynon 6/7/83 E. Handley E. Handley 6-7-83	
A	7/25/83	APPROVED Minor clerical changes		B. Solt R L Solt 7-21-83 S. Talwar A. Johnson R. Airhart 7/25/83

"THE INFORMATION CONTAINED IN THIS DOCUMENT IS CONFIDENTIAL AND PROPRIETARY TO BURROUGHS CORPORATION AND IS NOT TO BE DISCLOSED TO ANYONE OUTSIDE OF BURROUGHS CORPORATION WITHOUT THE PRIOR WRITTEN RELEASE FROM THE PATENT DIVISION OF BURROUGHS CORPORATION"

TABLE OF CONTENTS

TABLE OF CONTENTS

TABLE OF CONTENTS

PREFACE

This specification describes the MCP tables used by the B2000/B3000/B4000 Master Control Programs (MCPVI and MCPIX) and reflects changes through Release ASR 6.7.

REFERENCES

NUMBER	TITLE
1983 9927	MCP Program Interfaces

Features marked with an asterisk (*) are not available on Release ASR 6.7 of these MCPs but are being considered for some future release.

1 GENERAL DESCRIPTION

1.1 PURPOSE

1.2 PRODUCT CHANGES

1.2.1 ASR 6.7 Release Changes

OCS -- Operator Control Station
Address Block
Available Disk Table
Data Base Control Structure Table
Disk Directory Header Block
Disk File Header
File Header in Memory
DFHDR (Device Alternate)
Direct I/O FIB
EU/PACK Table
File Information Block Layout
Insert File
IOAT Layout
Job Reference Table
Loader Parameters
Memory Allocation Table
MIX Table
Maintenance Log Record
DCP Output Header Buffer
Disk Pack File Header
Time Sharing Language Processor Information Block
Pack Disk Label
Time Sharing Process Stack
Port Block
Queue Body Table
Run Log Record

1.2.1 ASR 6.7 Release Changes (Continued)

Time Sharing MIX Table
File Layout
Control Record Format

2 HARDWARE CODES

The terms hardware code or hardware type refer to the encoding of the hardware device name to a numeric value. These values are used in such records as the File Information Block and the Device Assignment Table. The following table contains the hardware codes and their mnemonic names.

01	CRD	Card Reader
02	PRN	Printer
03	PCH	Card Punch
04	MTP	Magnetic Tape
06	DSK	HPT or 100-Byte Mode Disk
07	NST	Non Status Device - Direct IO
08	SCR, S4A, S4B	Sorter/Reader
09	PTR	Paper Tape Reader
10	PTP	Paper Tape Punch
11	DPK	Disk Pack
13	TYP	Teletypewriter
16	TWX	TWX
20	B47, B35, B25	B2000/B3000/B4000 System
22	VDD	Visual Display Unit B9352
23	RJE	Remote Job Entry Terminal
25	BTT	Burroughs Touch Tone
26	BDD	Burroughs Digital Display Unit
28	OCS/SPO/TC4/ODT	OCS
29	TC7	TC 700
30	TC5	TC 500
31	B05	B 500
39	DCP/ISC/FEP	B 774/B 874
40	PBD	Printer Backup Disk
41	PCD	Punch Backup Disk
42	PBT	Printer Backup Tape
43	PBTB	Blocked Printer Backup Tape
44	PBP	Printer Backup Disk Pack
45	PCP	Punch Backup Disk Pack
50	PCR	Pseudo Card Reader, Disk (RLOG)
51	PCRP	Pseudo Card Reader, Disk Pack (RLOG)

3 MCP TABLES

The MCP tables presented in this specification represent the data areas maintained by the MCP. Much of this information can be found in a program memory dump and is presented for this reason.

The following tables are the ones which relate most directly to a program.

3 MCP TABLES (Continued)

The Disk/Disk Pack File Header	DF-=== or PF-===
The File Information Block	FIB===
The File Buffer Descriptor	FIB===
The Input/Output Assignment Table	IO-===
File Labels	LAB=== or USA===
The MIX Table	MIX-==
The Security Attributes Storage Area	SA-===

FUNCTION RECORD ACF-==

The FUNCTION RECORD is part of the USERFL and is used by system access control to determine the functions a user may legitimately perform.

ACF-ID	0-19	10 A	FUNCTION LIST ID (10 UA)
ACF-US	20-23	4 N	NUMBER OF USERS
ACF-FG	24	1 N	:8 REMOVE ON NULL ACTIVITY
			:4 DO NOT REMOVE (MCP FUNCTION LIST)
			:2 RESERVED
			:1 RESERVED
ACF-LV	25	1 N	USER CAPABILITY LEVEL (SPO LEVEL)
ACF-SC	26	1 N	DEFAULT USER SECURITYCLASS
			1 = PRIVATE
			2 = PUBLIC
			4 = GUARDED
			8 = CONTROLLED
ACF-F1	27	1 N	:8 USER CAN DO LIBMAINT TO OTHER USER'S FILES
			:4 USER CAN DO DIRECT I/O
			:2 RESERVED
			:1 RESERVED
ACF-AP	28	1 N	DEFAULT APPLICATION NUMBER
ACF-F2	29	1 N	:8 USER CAN DIALIN TO SYSTEM
			:4 USER CAN DO COPY TO CTLD
			:2 USER IS CANDE PRIVILEGED USER
			:1 RESERVED
	30-63	34 N	RESERVED

MEDIA RECORD ACM-==

The MEDIA RECORD is part of the USERFL and is used by system access control to determine if a given user may validly access the system from the input device he is attempting to use.

ACM-ID	00-19	10	A	MEDIA LIST ID
ACM-US	20-23	4	N	NUMBER OF USERS
ACM-FG	24	1	N	:8 REMOVE ON NULL ACTIVITY
				:4 DO NOT REMOVE (MCP LIST ENTRY)
				:2 "NOT" FLAG - MEDIA MATCH IS INVALID
				:1
ACM-PN	25	1	N	:8 PROCESSOR 3
				:4 PROCESSOR 2
				:2 PROCESSOR 1
				:1 PROCESSOR 0
ACM-ST	26-37	6	A	SITE ID
ACM-UT	38-49	6	A	UNIT OR STATION ID
ACM-HW	50-53	4	N	HARDWARE TYPE
ACM-F1	54	1	N	:8 STATION ID SUPPLIED
				:4 UNIT ID SUPPLIED
				:2 HARDWARE TYPE SUPPLIED
				:1 CC/U SUPPLIED
ACM-UN	55	1	N	UNIT NUMBER
ACM-CH	56-57	2	N	CHANNEL NUMBER
	58-63	6	N	

OCS -- OPERATOR CONTROL STATION AD-===

ADMASK is a copy of the IOAT Parameters. AD-PAR through AD-RES are expanded parameters for ease of MCP usage while in memory. A 4 KD buffer is required for each terminal that will be used for automatic table display. The buffer location is stored in IO-FIB. The first part of the buffer is used for control information. The remaining area is used for I/O data.

AD-OFL	0- 15	16	N	DISP OFLOW DISK ADDRESS
AD-BAS	16- 19	4	N	BUFFER BASE IN KD
AD-RSL	20- 23	4	N	RESULT DESCRIPTOR
AD-OPC	24- 25	2	N	OP CODE
AD-ADP	26- 27	2	N	ADAPTER
AD-VAR	28- 29	2	N	VARIANTS
AD-BAD	30- 35	6	N	BEGINNING ADDRESS
AD-EAD	36- 41	6	N	ENDING ADDRESS
AD-WAK	42- 46	5	N	WAKE UP TIME
AD-IOA	47- 53	7	N	IOAT INDEX
AD-INX	54- 55	2	N	INDEX INTO PARAMS
				INDEX TO NEXT PHYSICAL I/O TYPE
AD-WCH	56	1	N	LAST XMISSION TYPE
AD-STP	57	1	N	STOP-START FLAG
				:8 STOP DISPLAY
OC-OCT	58- 61	4	N	4 DIGIT SFO OVERFLOW
AD-OCT	58- 60	3	N	OVERFLOW TALLY
AD-ONX	61- 63	3	N	NEXT OFLOW TO READ
				TABLE TYPE CURRENTLY BEING BUILT OR DISPLAYED
ADMASK	64	6	N	CURRENT DISPLAY MASK
ADPREN	64	1	N	CURRENT PAREN FLAG
ADTIME	65- 66	2	N	CURRENT DISPLAY TIME
ADLINE	67- 68	2	N	CURRENT NO. OF LINES
ADTYPE	69	1	N	CURRENT TYPE
				GENERAL DISPLAY INFO AND FOR DISK SETUP
ADMESS	70	63	N	
AD-FLG	70	1	N	
				:8 <<AVAILABLE>>
				:4 <<AVAILABLE>>
				:2 PAGE OVERFLOW ALLOWED
				:1 "AD HDR" SPECIFIED
AD-SKP	71	1	N	NUMBER OF LINES TO SKIP BEFORE ACTIVE SCREEN AREA
AD-TAB	72	1	N	# OF TAB CHARACTERS BEFORE EACH TABLE LINE

```
          OCS -- OPERATOR CONTROL STATION                AD-===
                                                     (Continued)

                                FIRST OF UP TO SEVEN TABLE DISPLAY ENTRIES
AD-MSK          73    6  N   WHOLE MASK
AD-PAR          73    1  N   PARENTHESIS FLAG
AD-TIM          74    2  N   DISPLAY TIME
AD-LYN          76    2  N   LINES TO DISPLAY
AD-TYP          78    1  N   DISPLAY TYPE
                79   54  N   NINE MORE MASKS
AD-TAL         133    3  N   # OF OUTPUT LINES
AD-NXO         136    6  N   NEXT DATA GOES HERE
AD-TOT         142    2  N   LINE COUNT
AD-DTL         144    2  N   LINES MINUS SKIPS
AD-PRE         146    1  N   PARENTHESIS FLAG
AD-HED         147    1  N   1=HEADING PRINTED
AD-HDS         148    1  N   IO-HDS COPY
AD-SID         149    1  N   1=RIGHT SIDE AVAILABLE
AD-FRS         150    6  N   1ST DATA LINE HERE
AD-WUT         156    2  N   VARIABLE WAKE UP
ADCLRF         158    1  N   2=DQ MESSAGE OVERFLOW
               159    1  N   <<AVAILABLE>>AD-DTA
160-44391920 A  DATA TRANSMIT BUFFER

                                DEFINITION OF FIXED OUTPUT DATA
AD*SOH  160- 177    9  A   SOH/AD1/AD2/STX/ETC
AD*BEP  178- 181    2  A   ESC/? (BELL FOR TD830)
AD*HDR  182- 341   80  A   HEADER LINE
        342-44391829  A   REST OF BUFFER
```

ADDRESS BLOCK ADR-==

The disk area address block contains the area addresses of a HPT disk or disk pack file. The in-memory address blocks are kept for all random access files and for sequential disk pack files. (They are also kept for sequential disk files on MCPIX only.) Address blocks are not kept for device alternate files. The in-memory header of all files with address blocks will contain the address of this address block. The size of the block is 1 or 2 KD, depending on the number of areas declared for the file.

ADRBLK	0-1599	1600	N	IN-MEMORY ADDRESS BLOCK
ADRDSK	0- 15	16	N	DISK ADDRESS OF AREA # 1
	16-1599	1584	N	DISK ADDRESSES OF AREAS #2 - #100

OCS - INPUT BUFFER AI-===

There is one input buffer block allocated regardless of the number of ODT terminals declared on the system. It is used to process all input requests from the terminals and to assemble the messages in response to the input results.

AI-BUF		1000	N	BASE
AI-INX	0- 7	7	S	OUTPUT BUFFER POINTER
AI-OUT	8-167	80	A	CURRENT OUTPUT LINE
AI-INP	168-997	415	A	INPUT TEXT AREA
AI-ETX	998-999	1	A	ETX INPUT TERMINATOR

AVAILABLE DISK TABLE AV-===

The Available Disk Table has 40 primary blocks which are accessed directly by disk EU number (0 thru 39). Each block is 1000 digits long (5 segments). If the number of available table entries overflows the primary block for an EU, extension blocks will be obtained and linked to the primary block (or last extension block). The available Disk Table is shared by all processors on a system. The disk address of the first primary block is found in the Halt/Load Parameter Table. Available disk table blocks are linked both forward and backward. The links are circular, as the last forward link points to the primary block, and the first previous link points to the last extension block.

AVLTBL	0-999			AVAILABLE DISK TABLE BLOCK
AV-NEU	0- 1	2	N	EU OF NEXT BLOCK
AV-NAD	2- 7	6	N	ADR OF NEXT BLOCK
AV-PEU	8- 9	2	N	EU OF PREV BLCCK
AV-PAD	10- 15	6	N	ADR OF PREV BLOCK
AV-CNT	16- 17	2	N	NUMBER OF ENTRIES IN USE IN THIS BLOCK
AV-HLD	18	1	N	:8 ASSIGN TO HIGH DISK ONLY
				:4 ASSIGN TO LOW DISK ONLY
				:2 RESERVED
				:1 RESERVED
AV-PRB	19	1	N	1 = PRIMARY BLOCK, 0 = EXTENSION BLOCK
AV-BEU	20- 21	2	N	BLOCK EU NUMBER
AV-BAD	22- 27	6	N	ADR OF BLOCK
AV-EU#	28- 29	2	N	DISK EU NUMBER FOR THIS BLOCK
	30- 33	4	N	RESERVED
AV-ADR	34- 40	7	N	ENTRY #1: STARTING DISK ADDRESS (NO EU)
AV-SIZ	41- 47	7	N	ENTRY #1: NUMBER OF DISK SEGMENTS AVAILABLE
	48-999			ENTRIES #2 THRU # 69
AV-END	1000			END OF AVAILABLE DISK TABLE BLOCK

```
           COMPLEX WAIT TABLE                                    CWT-==

   CWT-TP    0- 3     4 N    ENTRY TYPE
                                 0000 = TIME
                                 0001 = ODT INPUT
                                 0002 = WRITE-OK
                                 0003 = READ-OK
                                 0004 = CHANGE-EVENT
                                 0005 = READY-EVENT
  +CWT-FA    4- 9     6 N    FILE ADDRESS
  +CWT-MC    4-11     8 N    DCP MCS POINTER
  +CWT-SI   10-13     4 N    SUB-FILE INDEX
   CWT-NX   14-15     2 N    INDEX OF ITEM IN LIST


   CWT-PG    0-15            PROGRAM HEADER ENTRY
   CWT-FG       0     1 N    PROGRAM HEADER FLAG = @F@
   CWT-RL    1- 4     4 N    RUN LOG NUMBER
   CWT-GV    5-10     6 N    GIVING ADDRESS
            11-15     5 N    RESERVED
```

DATA BASE CONTROL STRUCTURE TABLE DBC-==

The Data Base Control Structure Table is used by the data base extension module to contain information about each data base user in the system. It is also used as a queue for user DMS function requests and data base program request completions.

Data Base Control Structure Table format:

DBC-MX	0- 1	2 N	USER MIX NUMBER
DBC-IV	2- 3	2 N	USER INVOCATION NUMEER
DBC-TR	4	1 N	TERMINATE FLAG 0 = RUNNING 1 = TERMINATED 2 = TERMINATE-DBP TO BE NOTIFIED 4 = TERMINATING
DBC-US	5	1 N	USER STATE 1 = IN-DBP 2 = IN-DBP-OPEN/CLOSE 3 = WAITING-IOC 4 = WAITING-OPEN 5 = WAITING-CLOSE 6 = WAITING-FUNCTION-COMPLETE 7 = WAITING-TRANSFER-DATA 8 = NEW USER 9 = WAITING-DBP A = WAITING-IN-Q B = IOC C = OPEN-COMPLETE D = CLOSE-COMPLETE E = TRANSFER DATA COMPLETE F = RUNNING
DBC-OM	6	1 N	OPEN MODE 0 = CLOSED 1 = OPENED
DBC-FM	7	1 N	FUNCTION MODE 0 = NO FUNCTION REQUEST 1 = FUNCTION REQUEST OUTSTANDING 2 = DS THE USER
DBC-UR	8-13	6 N	ADDRESS OF DMERROR USE PROCEDURE
DBC-RG	14-19	6 N	ADDRESS OF DMSTATUS REGISTER
DBC-RA	20-25	6 N	ADDRESS OF RECORD AREA
DBC-SS	26-31	6 N	ADDRESS OF SET SELECTION STRING
DBC-SP	32-45	14 N	STORAGE AREA FOR STOPPED PROGRAMS
DBC-UP	46-52	7 N	USE PROCEDURE RETURN ADDRESS
DBC-TS	53-54	2 N	DBP STATE TABLE SLOT
DBC-RL	55-56	2 N	NEXT USER REQUEST LINK
DBC-PR	57-58	2 N	PREVIOUS USER REQUEST LINK
DBC-RC	59-60	2 N	NEXT DBP REQUEST COMPLETE LINK
DBC-PC	61-62	2 N	PREVIOUS REQUEST COMPLETE LINK

DATA BASE CONTROL STRUCTURE TABLE DBC-==
(Continued)

DBC-RR	63-64	2 N	DBP REQUEST RESULT - FILLED IN BY OPEN AND CLOSE
DBC-EM	65-66	2 N	DBP REQUEST RESULT - FILLED IN BY EXTENSION MODULE
DBC-UT	67	1 N	USER TYPE 0 = USER 1 = DBP
DBC-CM	68	1 N	CLOSE MODE FLAG 0 = NOT CLOSED 1 = CLOSE IN-PROGRESS 2 = RECOVERY/REORGANIZATION RUNNING
DBC-GO	69	1 N	USER STOP-GO FLAG 1 = SET BY EXT MOD-STOP PROG RUNNING 2 = SET BY MCP-STOP PROG RUNNING 4 = SET BY MCP-PROGRAM STOPPED
DBC-IU	70	1 N	DBC ENTRY IN USE
DBC-RP	71	1 N	USER PRIORITY AT BEGINNING OF FUNCTION
DBC-UU	72-75	4 N	AVAILABLE

DATA BASE PROGRAM (DBP) STATE TABLE DBP-==

The DBP State Table is used by the data base extension module to contain information about each DBP in the system. This table also contains the queue linkage for users that want to interface with a particular DBP and the queue linkage for function request completions for a particular DBP.

DBP STATE TABLE FORMAT

DBP-MX	0- 1	2	N	DBP MIX NUMBER
DBP-TR	2	1	N	TERMINATE FLAG 0 = RUNNING 1 = TERMINATE 2 = HOLD-TERMINATING 4 = TERMINATING
DBP-PS	3	1	N	DBP STATE 0 = RUNNING 1 = WAITING-BOJ 2 = NOT-RUNNING 4 = TERMINATED 8 = REFIRE BOJ
DBP-PA	4- 9	6	N	ADDRESS OF DBP PARAMETER AREA
DBP-NM	10-21	6	A	DATA BASE PROGRAM NAME
DBP-FU	22-23	2	N	LINK TO FIRST USER REQUEST
DBP-LU	24-25	2	N	LINK TO LAST USER REQUEST
DBP-FC	26-27	2	N	LINK TO FIRST REQUEST COMPLETE
DBP-LC	28-29	2	N	LINK TO LAST REQUEST COMPLETE
DBP-TS	30-31	2	N	SLOT # IN DBC FOR THIS DBP
DBP-UC	32-33	2	N	NUMBER OF USERS IN DBP
DBP-UO	34-35	2	N	NUMBER OF USERS OF DBP
DBP-GO	36	1	N	DBP STOP-GO FLAG 1 = SET BY EXT MOD-STOP PROG RUNNING 2 = SET BY MCP-STOP PROG RUNNING 4 = SET BY MCP-PROGRAM STOPPED
DBP-IU	37	1	N	DBP IN-USE FLAG 1 = NORMAL MODE 2 = RECOVERY/REORGANIZATION RUNNING
DBP-ST	38-39	2	N	DBP STATUS COUNT
DBP-IO	40-42	3	N	OUTSTANDING I/O COUNT
DBP-ER	43	1	N	I/O ERROR FLAG

DISK DIRECTORY HEADER BLOCK DDH-==

The disk directory consists of ten primary Disk Directory Header Blocks, one for each scramble string. The disk address of the first block is found in the Halt/Load Parameter Table. The Disk Directory Header Block is variable in length from 200 digits (12 entries, 1 segment) to 2800 digits (198 entries, 14 segments). The size is determined by Coldstart and is kept in the Halt/Load Parameter Table. When the number of permanent and temporary file entries exceeds the number of entries in the primary disk directory header block, extension blocks will be obtained and linked into the primary block (or last extension block). The Disk Directory Header Block is followed by the Disk Directory Security Block (if specified) and one disk file header segment for each entry in the Disk Directory Header.

```
DDHDR     0-2800         DISK DIRECTORY HEADER BLOCK (VARIABLE SIZE)
DDH-LE    0-   1   2 N   EU # OF EXTENSION BLOCK
DDH-LA    2-   7   6 N   ADDR OF EXTENSION BLOCK
DDH-FE    8-   9   2 N   EU # OF FIRST DISK FILE HEADER
DDH-FA   10-  15   6 N   ADDR OF FIRST DISK FILE HEADER
DDH-CT   16-  18   3 N   NUMBER OF ENTRIES IN USE IN THIS BLOCK
DDH-RM        19   1 N   0 = EXTENSION BLOCK, 1 = PRIMARY BLOCK
DDH-IE   20-  21   2 N   EU # OF DDHDR
DDH-IA   22-  27   6 N   ADDRESS OF DDHDR
DDH-ID   28-  39   6 A   ENTRY #1:  DISK FILE ID
DDH-ST        40   1 N   ENTRY #1:  DISK FILE STATUS DIGIT
                         :8 ENTRY IS IN USE
                         :4  SET = PERMANENT FILE, RESET = TEMP FILE
                         :2  FILE SECURITY IS USED
                         :1  NO LIBRARY MAINTENANCE (PERMANENT FILES)
                         :1  REMOVE ON NULL ACTIVITY (TEMP FILES)
DDH-LK        41   1 N   ENTRY #1:  LOCK STATUS DIGIT
                         :8 NEXT ENTRY IS CONTINUATION OF FILE
                         :421 LOCK STATUS (PERMANENT FILES)
                         000 = NO ACTIVITY
                         100 = READ ONLY ACTIVITY
                         010 = WRITE ACTIVITY (& POSSIBLE READ)
                         001 = LOCK ACCESS (& POSSIBLE READ)
                         111 = LOCK NO ACCESS
                         :4 MAKE PERMANENT FILE ON HALT/LOAD
                            (TEMP FILE)
                         :21  PROC # OF FILE CREATOR  (TEMP FILES)
         42-2799         ENTRIES #2 THRU #198
```

DISK FILE SECURITY DIRECTORY DDS-==

When the security option is set, the MCP will get additional disk segments for the Disk File Security Directory. Its position is between the Disk Directory Header Block and the Disk File Header Block. There are as many entries as there are slots for file ids in the Disk Directory Header Block. Each segment has space for 5 twenty byte entries. The segment and position in the segment must be calculated for each file. This allows a one segment disk read.

	0-199			ONE SEGMENT OF DIRECTORY. REPEAT FOR ADDITIONAL FILES.
DDS-TY	0	1	N	SECURITYTYPE ATTRIBUTE 8=CONTROLLED 4=GUARDED 2=PUBLIC 1=PRIVATE
DDS-SU	1	1	N	SECURITYUSE 6=IO 4=IN 2=OUT 1=SECURED
DDS-SN	2	1	N	SENSITIVEDATA ATTRIBUTE 1=OVERWRITE DATA WHEN FILE IS REMOVED
DDS-MA	3	1	N	SECURITYMAINT (NOT IMPLEMENTED) :8=ADD :4=DUMP :2=CHANGE :1=REMOVE
DDS-UC	4- 23	10	A	USERCODE OF THE FILE CREATOR.
DDS-GF	24- 35	6	A	GUARDFILE ID
	36- 39	4	N	RESERVED
	40-199			4 MORE ENTRIES.

DISK FILE SECURITY DIRECTORY DDS-==
(Continued)

Disk File Headers

A file that resides on disk or disk pack has three components. Two of these are maintained by the MCP: Disk Directory entry and file header. The third, file data, is user program created and altered.

The Disk Directory entry simply indicates that a particular file resides on disk or disk pack. Presence of a Disk Directory entry implies the existence of a File Header and possibly, file data.

The File Header contains information about the file and acts as a road map for the file data. Pointers to the allocated areas of the file (pages) are in the File Header. Also contained in the header are record length, blocking factor, number of areas declared, number of users of the file, and various other information about the file.

Different information is maintained in the File Headers for disk files than for disk pack files. However, after a file is OPENed, file processing requires similar information whether the file is on disk or disk pack. Consequently, when a file is OPENed, the two file header types are reformatted into a common description when the header is loaded into memory.

A user program can request disk file header information in two ways. 1) the INTERROGATE FILE BCT; 2) load file header information at file OPEN time (FIBST1 set). Refer to Program Interface PS #1983 9927 and FIBs in this section, for details.

A disk/disk pack file OPEN requires at least 200 digits of memory. 100 digits is allocated to an IOAT, the other 100 is the reformatted file header (refer to IOAT in this section).

The area pointers to the file (address blocks) are allocated in 1 or 2 KD blocks. Area pointers are sixteen digits long. Thus if the file has less than 63 areas, 1 KD is required for a disk/disk pack file open; if it has 63 or more areas declared, 2 KD is required.

The following descriptions are for the headers on disk (for disk pack, see PF-===), as well as the reformatted header in memory after the file has been OPENed.

Disk File Header (on disk) DF-===

The following descriptions apply to the disk version of the Disk File Header which resides in the disk directory blocks.

DF-RSZ	0- 4	5	N	Record size in digits
DF-RPB	5- 7	3	N	Maximum number of records per block
DF-#AR	8- 9	2	N	Maximum number of disk areas assigned to the file
DF-EOF	10- 17	8	N	End of file pointer
DF-USE	18- 19	2	N	Number of users on processor 0
	20- 21	2	N	Number of users on processor 1
	22- 23	2	N	Number of users on processor 2
	24- 25	2	N	Number of users on processor 3
DF-SYS	26	1	N	System number of locking program
DF-MIX	27- 28	2	N	Mix number of locking program
DF-ORG	29	1	N	:8 RESERVED :4 INDEXED I/O DATA FILE :2 INDEXED I/O KEY FILE :1 RELATIVE I/O DATA FILE
	30	1	N	RESERVED
DF-ST1	31	1	N	:8 Get high disk areas :8/ Get low disk areas :4 SET = Inhibit APCR and APBD options for this file. "RN=", "PBD=", "PC=", and "PM=" commands inhibited for this file. :2 Remove on HL even if marked permanent :1 Do not squash file
DF-DKS	32	1	N	Disk subsystem assignment 0 = Default disk subsystems 1 = Primary disk subsystem 2 = Disk subsystem # 2 3 = Disk subsystem # 3 4 = Disk subsystem # 4 5 = Disk subsystem # 5 6 = Disk subsystem # 6 7 = Disk subsystem # 7 8 = Disk subsystem # 8 E = Common (shared) disk subsystems
DF-DSA	33- 39	7	N	Number of disk segments per disk area
DF-EU1	40- 41	2	N	EU # of first area of file
DF-AD1	42- 47	6	N	Address of first area of file
	48-839	792	N	Disk address of areas # 2 through # 100

FILE HEADER IN MEMORY DF-===

The following information represents the reformatted disk and disk pack file headers when the headers are in memory.

DF-RSZ	0-	4	5	N	Record size in digits
DF-RPB	5-	7	3	N	Number of records per block
DF-#AR	8-	9	2	N	Maximum number of areas
DF-EOF	10-	17	8	N	End of file pointer
+DF-ORS	18-	22	5	N	Original record size in digits (all but split cylinder)
+DF-PPA	18-	21	4	N	Partitions per area (split cylinder disk)
+DF-BPP	22-	25	4	N	Blocks per partition (split cylinder disk)
+DF-ORB	23-	25	3	N	Original number of records per block (all but split cylinder)
DF-DSB	26-	29	4	N	Number of disk segments per block or pack sectors per block
DF-PAK		30	1	N	File assignment type :8 Assign by space available pack file :4 Assign by area pack file :2 Single pack :2/ Multipack file :1 Cylinder bound pack file
DF-ST1		31	1	N	:8 Get high disk address :8/ Get low disk address :4 SET = Inhibit APCR and APBD options for this file. "RN=", "PBD=", "PC=", and "PM=" commands inhibited for this file. :2 Remove on HL even if marked permanent :1 Do not squash file
DF-DKS		32	1	N	Disk subsystem assignment 0 = Default disk subsystems 1 = Primary disk subsystem 2 = Disk subsystem #2 3 = Disk subsystem #3 4 = Disk Subsystem #4 5 = Disk Subsystem #5 6 = Disk Subsystem #6 7 = Disk Subsystem #7 8 = Disk Subsystem #8 E = Common (shared) disk subsystem
DF-DSA	33-	39	7	N	Number of disk segments per disk area
DF-ADR	40-	47	7	S	Memory address of address block
DF-DIR	48-	63	16	N	Disk address of Disk Directory header block or pack address of Pack Directory sector
DF-DFH	64-	79	16	N	Disk address of disk file header block or pack address of pack file header
DF-DRX	80-	83	4	N	Index to file in Disk Directory header block or pack directoy header sector

FILE HEADER IN MEMORY DF-===
(Continued)

DF-BPA	84- 90	7	N	Number of blocks per disk area
DF-SIZ	91- 93	3	N	Disk file header size in digits (840 digits max) or Pack file header size in bytes (500 bytes max)
DF-USR	94- 95	2	N	Total number of users sharing in-memory DFHDR
DF-RND	96- 97	2	N	Total number of users with random access
DF-ST2	98	1	N	:8 In-memory area address blocks present :4 Temporary disk or pack file :2 Reserved :1 Reserved
DF-OR1	99	1	N	:8 Reserved :4 Indexed I/O Data File :2 Indexed I/O Key File :1 Relative I/O Data File

DFHDR (DEVICE ALTERNATE)

This version of the Disk File Header resides in memory as a 124 digit block which is virtually identical to the normal In-Memory Disk File Header above. This version of the DFHDR applies only to device alternate files (pseudo reader input files, printer/punch backup disk output files, and blocked printer backup tape files). Since in all of these cases the DFHDR resides in a Type 4 block immediately above an IOAT and below the actual buffer, we are able to extend the header by 24 digits to contain information about the buffer status.

DF-HDR	0-123			IN-MEMORY DISK FILE HEADER (DEVICE ALTERNATE)
DF-RSZ	0- 4	5	N	Record size in digits
DF-RPB	5- 7	3	N	Number of records per block
DF-#AR	8- 9	2	N	Maximum number of areas
DF-EOF	10- 17	8	N	End of file pointer
+DF-ORS	18- 22	5	N	Original record size in digits (all but split cylinder)
+DF-PPA	18- 21	4	N	Partitions per area (split cylinder disk)
+DF-BPP	22- 25	4	N	Blocks per partition (split cylinder disk)
+DF-ORB	23- 25	3	N	Original number of records per block (all but split cylinder)
DF-OSB	26- 29	4	N	Number of disk segments per block or pack sectors per block
DF-PAK	30	1	N	File assignment type :8 Assign by space available pack file :4 Assign by area pack file :2 Single pack :2/ Multipack file :1 Cylinder bound pack file
DF-ST1	31	1	N	 :8 Get high disk address :8/ Get low disk address :4 SET = Inhibit APCR and APBD options for this file. "RN=", "PBD=", "PC=", and "PM=" commands inhibited for this file. :2 Remove on HL even if marked permanent :1 Do not squash file
DF-DKS	32	1	N	Disk subsystem assignment 0 = Default disk subsystems 1 = Primary disk subsystem 2 = Disk subsystem #2 3 = Disk subsystem #3 4 = Disk Subsystem #4 5 = Disk Subsystem #5 6 = Disk Subsystem #6 7 = Disk Subsystem #7 8 = Disk Subsystem #8

DFHDR (DEVICE ALTERNATE) (Continued)

				E = Common (shared) disk subsystem
DF-DSA	33- 39	7	N	Number of disk segments per disk area
DF-ADR	40- 47	7	S	Memory address of address block
DF-DIR	48- 63	16	N	Disk address of Disk Directory header block or pack address of Pack Directory sector
DF-DFH	64- 79	16	N	Disk address of disk file header block or pack address of pack file header
DF-DRX	80- 83	4	N	Index to file in Disk Directory header block or pack directoy header sector
DF-BPA	84- 90	7	N	Number of blocks per disk area
DF-SIZ	91- 93	3	N	Disk file header size in digits (840 digits max) or Pack file header size in bytes (500 bytes max)
DF-USR	94- 95	2	N	Total number of users sharing in-memory DFHDR
DF-RND	96- 97	2	N	Total number of users with random access
DF-ST2	98	1	N	
				:8 In-memory area address blocks present
				:4 Temporary disk or pack file
				:2 Reserved
				:1 Reserved
DF-OR1	99	1	N	:8 Reserved
				:4 Indexed I/O Data File
				:2 Indexed I/O Key File
				:1 Relative I/O Data File
DF-CRX	100-103	4	N	INDEX TO CURRENT RECORD IN BUFFER
DF-BSD	104-107	4	N	BUFFER SIZE IN DIGITS
DF-BRS	108-110	3	N	BUFFER RECORD SIZE IN DIGITS
DF-CBS	111-112	2	N	MEMORY BLOCK SIZE IN K DIGITS
DF-URS	113-114	2	N	USER RECORD SIZE IN WORDS
DF-PRX	115-116	2	N	PSEUDO READER DIRECTORY INDEX
DF-RLT	117	1	N	LAST DIGIT OF PSEUDO READER RESULT DESCRIPTOR
DF-DSC	118-123	6	N	FIRST SIX DIGITS OF I/O DESCRIPTOR (A ADDRESS IS DF-CRX; B ADDRESS IS DF-BSD; THE FOUR HIGH-ORDER DIGITS OF THE IOAT POINTER ARE USED AS THE I/O BASE ADDRESS).
DF-BFR	124-???			DEVICE ALTERNATE BUFFER

DIRECT I/O FIB DIR-==

DIRFIB	0- 59			DIRECT I/O FILE INFORMATION BLOCK
DIR-AC	0- 1	2	N	SECONDARY I/O CHANNEL
DIR-PC	2- 3	2	N	PRIMARY I/O CHANNEL
DIR-UN	4	1	N	HARDWARE UNIT NUMBER
DIR-ST	5	1	N	FILE STATUS DIGIT
				0 = FILE IS OPEN
				1 = FILE IS CLOSED
				2 = MAP IOAT
DIR-S1	6	1	N	:8 INHIBIT CHANNEL AT CLOSE TIME
				:4 INHIBIT (XU) UNIT AT CLOSE TIME
				:2 USE ALTERNATE CHANNEL
				:1 DISK I/O
DIR-XD	7	1	N	:8 BINARY ADDRESSES REQUIRED
				:4 DIRECT READ ("LOGICAL") FIB
				:2 100 DIGIT FIB FOR LCP USE
				:1 USE XD DISK AREA
DIR-RT	8- 9	2	N	ERROR RETRY COUNTER
DIR-IM	10- 13	4	N	ERROR IGNORE MASK
DIR-IO	14- 19	6	N	ABSOLUTE IOAT ADDRESS
DIR-RD	20- 23	4	N	RESULT DESCRIPTOR STORAGE AREA
	24- 47	4	N	I/O DESCRIPTOR
DIR-OP	24- 25	2	N	I/O DESCRIPTOR OP CODE
DIR-V1	26- 27	2	N	I/O DESCRIPTOR VARIANTS #1 & 2
DIR-V3	28	1	N	I/O DESCRIPTOR VARIANTS #3
DIR-V4	29	1	N	I/O DESCRIPTOR VARIANTS #4
DIR-AA	30- 35	6	N	I/O DESCRIPTOR A-ADDRESS
DIR-BB	36- 41	6	N	I/O DESCRIPTOR B-ADDRESS
DIR-AD	42- 47	6	N	I/O DESCRIPTOR DISK ADDRESS
DIR-R1	48- 53	6	N	"A" RAD ADDRESS
DIR-R2	54- 59	6	N	"B" RAD ADDRESS
DIR-EX	60- 75	6	N	FOUR WORDS OF DLP R/D
DIR-CN	76- 81	6	N	CANCEL OPCODE
DIR-EU	82- 84	3	N	LOGICAL EU NUMBER FOR XD/LOGICAL READ
	85- 87	3	N	RESERVED
DIR-LA	88- 99	12	N	DECIMAL ADDR FOR LOGICAL READ OR XD OPEN

DISK SUBSYSTEM TABLE DK-===

DK-LNK	0-	1	2	N	HEAD/TAIL TABLE LINK
DK-CHN	2-	3	2	N	PRIMARY I/O CHANNEL
DK-FRM	4-	7	4	N	FIRMWARE LEVEL FROM READ-UNIT-ID OP
DK-ID	8-	19	6	A	FIRMWARE FILE ID ON DISK
DK-BMD	20-	25	6	N	BUFFER MEMORY DUMP OPCODE
DK-POP	26-	31	6	N	QUEUED SUBSYSTEM POLL OPCODE
DK-QWK	32-	37	6	N	IN-LINE SUBSYSTEM POLL OPCODE
DK-POL	38-	41	4	N	SUBSYSTEM POLL MASK (SEEKING DRIVES)
DK-PCH	42-	43	2	N	CHANNEL OF QUEUED SUBSYSTEM POLL
DK-ADR	44-	49	6	N	ADDRESS OF SUBSYSTEM POLL RESULT
DK-EU#	50-	81	32	N	EU # ARRAY INDEXED BY PHYS. UNIT NUMBER
DK-SUB		82	1	N	PHYSICAL SUBSYSTEM LINK NUMBER (0 TO F)
DK-ST1		83	1	N	:8 FPM PRESENT ON SUBSYSTEM
					:4 1A/1C TYPE CONTROL (EU RANGE = 00 - 39)
					:2 LCP DISK CONTROL
					:1 FIRMWARE VALIDATION NOT REQUIRED
DK-ST2		84	1	N	:8 DK-FPM IS VALID (SUBSYSTEM IS SHARED)
					:4 RESERVED
					:2 FIRMWARE FILE NON-PRESENT OR INVALID
					:1 FIRMWARE FILE HAS BEEN VALIDATED
DK-FPM		85	1	N	PHYSICAL FPM SUBSYSTEM NUMBER
	86-	99	14	N	RESERVED

MESSAGE HEADER DP-===

DP-FUN	0- 1	2	N	FUNCTION CODE
DP-OFN	2- 3	2	N	ORIGINAL FUNCTION CODE
DP-LSN	4- 7	4	N	LOGICAL STATION NUMBER
DP-VAR	8- 11	4	N	FUNCTION VARIANTS
DP-TXS	12- 15	4	N	TEXT LENGTH IN BYTES
DP-ERR	16- 23	8	N	ERROR BOOLEANS
DP-LES	24- 25	2	N	LAST ERROR FLAG SET
DP-RTY	26- 27	2	N	RETRY COUNT
DP-TAL	28- 33	6	N	NDL TALLY FIELD
DP-TOG	34- 35	2	N	NDL TOGGLE FIELDS
+DP-TOP	36	1	N	:1 TOP QUEUE FLAG
+DP-TR#	36- 39	4	N	TRANSMISSION NUMBER
DP-RBI	40- 41	2	N	RESULT BYTE INDEX
DP-MCS	42- 43	2	N	MCS #
DP-MS#	44- 47	4	N	MESSAGE NUMBER
DP-SEQ	48- 55	8	N	SEQUENCE NUMBER
DPOVR	56- 59	4	N	ORIGINAL VARIANTS
DP-MPT	60- 63	4	N	DUMP FUNCTION S-MEMORY POINTER (Hexadecimal)
	64- 67	4	N	RESERVED
DP-SMP	68- 71	4	N	S-MEMORY POINTER FROM DCP (Hexadecimal)
DP-DC#	72	1	N	DCP TABLE INDEX
	73- 79	7	N	RESERVED

DISK PACK DIRECTORY DPD-==

DPDIR	0-359			DISK PACK DIRECTORY SECTOR
DPD-FL	0- 7	8	N	FORWARD LINK
DPD-BL	8- 15	8	N	BACKWARD LINK
DPD-SP	16- 23	8	N	ADDRESS OF THIS SECTOR
DPD-MK	24	1	N	MARKER
	25- 27	3	N	AVAILABLE
DPD-FG	28	1	N	"F" VALIDITY FLAG
	29	1	N	AVAILABLE
DPD-AH	30- 37	8	N	ADDRESS OF FILE HEADER
DPD-SZ	38- 41	4	N	LENGTH OF FILE HEADER IN BYTES
DPD-NM	42- 57	8	A	FILE NAME
DPD-VF	58	1	N	:8 BASE PACK
				:4 LOCK ON TERMINATE
				:2 TEMPORARY FILE
				:1 PERMANENT FILE
				:0 ENTRY AVAILABLE
DPD-ST	59	1	N	:8 OPEN LOCKOUT
				:4 OPEN LOCK ACCESS
				:2 OPEN OUTPUT
				:1 OPEN INPUT
				:0 INACTIVE
	60-359			10 MORE 30 DIGIT DIRECTORY ENTRIES

DUSTAT - EUSTAT

The following two arrays [DUSTAT, EUSTAT] are used to maintain the availability of peripheral units [DUSTAT] and disks and packs [EUSTAT]. Q-STAT in IO QUEUE elements contains a pointer to the unit entry for which they are queued. EUSTAT is indexed by EU number and DUSTAT is indexed by IO-STA value. Since an EU # of zero is invalid EUSTAT [00] is used for the FPMS. The value of the bits in each digit are as follows:

```
1:8 = UNIT INHIBITED
1:4 = <<AVAILABLE>>
1:2 = SEEK INITIATED (EUSTAT ONLY)
1:1 = I/O IN PROGRESS
```

ERROR TABLE

The following tables are used by the I/O Error routine in its analysis and handling of exception result descriptors.

The first table is used to locate the cases of result descriptors allowed for a given HARDWARE/OP code pair, ensure a valid result descriptor exists, and supply a fix code for handling the error.

Each 6 digit set either defines a HARDWARE/OP code pair or a result descriptor case code and error handling data. Two leading digits =@FF@ identify a HARDWARE/OP code pair and is followed by a 2 digit hardware code and a 2 digit OP code. A first digit of other than an undigit F identifies a result descriptor error case and implies the following R/D value:

00	=	110
01	=	001
02	=	002
03	=	004
04	=	008
05	=	010
06	=	020
07	=	040
08	=	080
09	=	100
0A	=	120
0B	=	130
0C	=	140
0D	=	180
0E	=	200
0F	=	030
10	=	013
11	=	F00
12	=	F80
13	=	F10
14	=	CA0
15	=	C18

The third and fourth digits provide Booleans for identifying standard error recovery procedures.

The third digit is defined as follows:

8 BIT = BACKUP REQUEST
4 BIT = REMOVE UNIT INHIBIT
2 BIT = REQUE ELEMENT ON ERROR OVERFLOW
1 BIT = I/O ERROR COUNT MAINTENANCE

ERROR TABLE (Continued)

The fourth digit is defined as follows:

8 BIT = EXIT TO TESTER AT COMPLETION
4 BIT = ALLOW UNRECOVERED ERROR ON MCP I/O
2 BIT = ACCUM DEVICE ERRORS
1 BIT = RECORD ERROR IN MAINTENANCE LOG

The fifth digit specifies the value which is placed in the Queue Element Result Descriptor Flag Q-RDFG (see I/O Queue Element definition for meaning).

The fifth and sixth digit combination identify the specific condition which resulted in the error descriptor case being handled. This value is used in Global and Overlayable I/O error routines to link to specific handling code for the error. The first digit of this field is moved into Q2RDFG upon IO complete. The allowable conditions are as follows:

00 = EXPECTED CONDITION - IGNORE ERROR
01 = UNIT REWINDING - INITIATED BY REWIND OP CODE
02 = LOW PAPER - I/O COMPLETED SUCCESSFULLY
03 = DATA ERROR CORRECTION - H.P.T. EMULATOR
10 = INVALID RESULT DESCRIPTOR - CASE UNDEFINED
20 = INVALID INITIATE I/O - HARDWARE DETECTED
21 = I/O DESCRIPTOR ENDING ADDRESS ABOVE LIMIT REGISTER
22 = DISK WRITE WITHIN MCP SEGMENTS
23 = INVALID DISK ADDRESS
24 = DISK WRITE/UNLOCK ON ADDRESS LOCKED BY OTHER USER
25 = NO RESULT DESCRIPTOR FROM CONTROL AFTER 10 SECONDS
26 = NON-PRESENT OPTION REQUIRED
27 = UNIT REWINDING - NON-REWIND OP CODE
28 = NO WRITE RING
29 = DISK WRITE LOCKOUT
2A = DISK NOT READY
2B = DISK TIME OUT
2C = DISK EU BUSY
2D = LCP CHANNEL FAILURE ILLEGAL STATUS OR TIMEOUT

EU/PACK TABLE

EU-===

EUTAB	0-199			DISK/DISKPACK DEVICE TABLE	
EU-HDW	0- 1	2	N	HARDWARE TYPE (06=DISK / 11=PACK)	
EU-TYP	2	1	N	DRIVE TYPE INDEX	
				0 = UNDEFINED	
				1 = HEAD-PER-TRACK DISK (1A, 1C, ETC.)	
				2 = LAK DISK (I.E. 235 THRU HTC1A)	
				3 = LCP DISK (I.E. 235 THRU HT-DLP)	
				4 = UIO 5N-DLP	
				5 = UIO HT-DLP	
				6 = RESERVED	
	3	1	N	RESERVED	
EU-CHN	4- 5	2	N	PRIMARY CHANNEL NUMBER	
EU-LNK	6- 7	2	N	HEAD/TAIL TABLE LINK	
EU-STA	8- 9	2	N	IO-STA EQUIVALENT	
EU-QUE	10- 11	2	N	I/O QUEUE ELEMENT COUNT	
EU-MIX	12- 13	2	N	DIRECT I/O MIX NUMBER	
EU-FIB	14- 19	6	N	DIRECT I/O RELATIVE FIB ADDRESS	
EU-UST	20- 22	3	N	TEST OP RESULT (UNIT STATUS)	
EU-SUB	23	1	N	PHYSICAL SUBSYSTEM NUMBER	
EU-SB#	24	1	N	LOGICAL DISK SUBSYSTEM NUMBER	
EU-ERC	25- 27	3	N	TOTAL ERROR COUNT ON DEVICE	
EU-ERT	28- 29	2	N	TOTAL RETRY COUNT PER FAILING I/O	
EU-BCT	30- 37	8	N	PHYSICAL I/O COUNT (BLOCK COUNT)	
EU-UN#	38- 41	4	N	PHYSICAL UNIT NUMBER MASK	
EU-UNC	42- 45	4	N	UNCONDITIONAL I/O VARIANT MASK	
EU-POL	46- 49	4	N	SUBSYSTEM POLL MASK FOR THIS UNIT	
EU-FAM	50- 61	6	A	DISKPACK NAME	
EU-HDS	62- 63	2	N	DRIVE TYPE (SUPPLEMENTARY HARDWARE TYPE)	
				00 = UNKNOWN TYPE	180 BYTE
				01 = 215 PACK	180 BYTE
				02 = 225 PACK	180 BYTE
				03 = 235 PACK	180 BYTE
				04 = 206 PACK (INTERLACED)	180 BYTE
				05 = 206 PACK (SEQUENTIAL)	180 BYTE
				06 = 207 PACK (INTERLACED)	180 BYTE
				07 = 207 PACK (SEQUENTIAL)	180 BYTE
				08 = 677 PACK (INTERLACED)	180 BYTE
				09 = 677 PACK (SEQUENTIAL)	180 BYTE
				0A = 659 PACK (INTERLACED)	180 BYTE
				0B = 659 PACK (SEQUENTIAL)	180 BYTE
				21 = DISK SYSTEMS MEMORY	100 BYTE
				22 = DISK A1	100 BYTE
				23 = DISK 1A-5	100 BYTE
				24 = DISK 1C-1	100 BYTE
				25 = DISK 1C-2	100 BYTE
				26 = DISK 225	100 BYTE
				27 = DISK 5N	100 BYTE
				28 = DISK 235	100 BYTE
				29 = DISK 206 (INTERLACED)	100 BYTE

EU/PACK TABLE

EU-===
(Continued)

				2A = DISK 206 (SEQUENTIAL) 100 BYTE
				2B = DISK 207 (INTERLACED) 100 BYTE
				2C = DISK 207 (SEQUENTIAL) 100 BYTE
				2D = DISK 677 (INTERLACED) 100 BYTE
				2E = DISK 677 (SEQUENTIAL) 100 BYTE
				2F = DISK 659 (INTERLACED) 100 BYTE
				30 = DISK 659 (SEQUENTIAL) 100 BYTE
EU-XXX	64	1	N	RESERVED FOR IO-ST1 POSITION
EU-LAB	65	1	N	DEVICE LABEL SECTOR ADDRESS (0 OR 4)
EU-UNT	66- 67	2	N	LOGICAL UNIT NUMBER (00 TO 39)
EU-CTR	68- 69	2	N	NUMBER OF SEEKING/BUSY R/D'S IN 10 SECONDS
EU-TST	70- 75	6	N	"TEST" OPCODE FOR THIS UNIT
EU-PSN	76- 81	6	N	MEDIA SERIAL NUMBER
EU-LOW	82- 93	12	N	LOWEST VALID ADDRESS (TOP OF MCP/LABEL AREA)
EU-FOC	94- 97	4	N	NUMBER OF FILE AREAS OPEN ON DRIVE
EU-TAS	98-109	12	N	TOTAL AVAILABLE SECTORS ON DRIVE
EU-FAS	110-121	12	N	ADDRESS OF AVAILABLE TABLE (PACK)
EU-FDS	122-133	12	N	ADDRESS OF DIRECTORY (PACK)
EU-TIM	134-137	4	N	I/O START TIME IN MS.
EU-IOT	138-143	6	N	ACCUMULATED I/O TIME THIS STATUS PERIOD
EU-AV	144-147	4	N	AVERAGE I/O UTILIZATION IN PERCENT (999V9)
EU-LKS	148	1	N	:8 DRIVE IS SAVED
				:4 DRIVE IS TO BE SAVED
				:2 DRIVE IS WAITING POWER OFF
				:1 RESERVED
EU-NSC	149	1	N	:8 DRIVE IS DEAD (CONTROLLER FAILURE, ETC.)
				:4 DRIVE IS NOT READY (OFFLINE, ETC.)
				:2 DRIVE IS WAITING NOT READY
				:1 RESERVED
EU-ST1	150	1	N	:8 FORCE I/O'S UNCONDITIONAL
				:4 DRIVE NEEDS 100 SECOND STATUS
				:2 UNIT IS DLP
				:1 DRIVE DECLARED READONLY
EU-ST2	151	1	N	:8 FOREIGN PACK
				:4 DRIVE IS WRITE LOCKED OUT
				:2 RESTRICTED PACK
				:1 MASTER PACK
EU-SB1	152	1	N	:8 FPM PRESENT ON PHYSICAL SUBSYSTEM
				:4 LOGICAL SUBSYSTEM IS DEFAULT DISK
				:2 RESERVED
				:1 DEVICE IS SHARED
EU-SB2	153	1	N	:8 ASSIGN TEMP FILES TO HIGH DISK
				:4 ASSIGN TEMP FILES TO LOW DISK
				:2 RESERVED
				:1 RESERVED
EU-FPM	154	1	N	PHYSICAL FPM SUBSYSTEM NUMBER
EU-BIN	155	1	N	BINARY ADDRESSES REQUIRED
EU-EU#	156-158	3	N	LOGICAL EU NUMBER

```
EU/PACK TABLE                                          EU-===
                                                        (Continued)

159-199  41 N   RESERVED
```

FILE INFORMATION BLOCK FIB===

When a program is compiled, each source language file declaration causes the compiler to generate a block of data which describes the file for the MCP. This 200-digit area is known as the File Information Block (FIB) and contains flags, counters, addresses, and sizes which are used by the MCP during all phases of file processing.

Many of the fields are directly constructed by the compiler according to the source program declaration. These include record size, blocking factor, file hardware type, recording mode, label convention, and so on. Other fields affected when the file is OPENed are input/output mode, file status flag, and link to a physical device table entry. Several fields are dynamically changed during file processing such as the block and record counts; record and buffer pointers.

Several fields are used only for certain types of files; other fields have multiple uses depending on the file type or the particular stage of file processing.

Any modification of the FIB is emphatically discouraged. Such actions can be dangerous to the program attempting the modification because the information can cause MCP or program failure. Further, as MCP mechanisms are subject to change, there is no guarantee that programs which modify FIBs will continue to function properly under future releases. The information in the following sub-sections is provided primarily to give greater insight into the FIB-MCP interface. To a lesser degree, the information is provided for those who are willing to risk the dangers inherent in FIB modifications. The release of such information does not constitute endorsement of its application, does not imply support of its use, and does not guarantee that programs which employ these or similar mechanisms will continue to function.

The FIB is the file handling interface between program and MCP. The 200-digit area is used for READ/WRITE, OPEN, and CLOSE. The following descriptions indicate the functions of the FIB fields. Note that some fields have different meanings depending on hardware type.

```
          FILE INFORMATION BLOCK                    FIB===
                                                      (Continued)

     FIBST1         0 1 N     Controls tape assignment, DFH access, and
                              file recovery
                            MTP OPEN OUTPUT:
                                If none of FIBST1:8, FIBST1:4, and
                                FIBOPT:4 are set; all three bits are
                                set.  If FIBST1:8 is set, but FIBOPT:4
                                is not set, FIBOPT:4 is set.  At the
                                end of file OPEN the type of tape
                                actually assigned reflects the bits
                                left on in FIBST1, FIBOPT.
                            :1  DSK (input, I/O):
                                At file OPEN, MCP puts first 40 digits
                                of DFH as exist in directory into
                                requestors BASE:+100.
                                See FIBST1:8.
                                DSK (output):
                                At file CLOSE, requestor wants MCP to
                                use 18 digits at BASE:+100 rather than
                                MCP maintained DFH.
                            :2  DSK, MTP (output):
                                File is CLOSEd with RELEASE if not done
                                by program prior to EOJ (normal or
                                abnormal).
                            :4 DSK:
                                At file OPEN, MCP puts first 40 digits
                                of DFH as exist in memory into
                                requestors BASE:+100 (includes any
                                modification made to accommodate
                                programmatically declared redefinition
                                in block size, and so on).
                            :4  MTP:
                                Specifies MT7 acceptable for file.
                            :8  DSK (input, I/O):
                                At file OPEN, MCP puts disk addresses
                                for file starting at BASE:+140;
                                number of addresses passed depends on
                                number allowed for file as specified
                                in DFH; bit is reset during file OPEN
                                procedure.
                            :8  MTP:
                                Specifies MT9 (NRZ) acceptable.
                            :8  SOR:
                                Specifies 4A type Sorter Control
                                required at OPEN.
```

FILE INFORMATION BLOCK FIB===
(Continued)

```
FIBST2        1 1 N    Controls Printer/Punch assignments;
                       miscellaneous.
                    :1   Input:
                         Indicates no physical file to process
                         or CLOSE; set if OF or FR keyboard
                         message entered, all backup files
                         accessed (for example, @00000
                         request) or last file on multifile tape
                         accessed (FID = spaces request).
                    :2   PRN, PCH:
                         File must be assigned to primary
                         device.
                    :4   PRN, PCH:
                         File must be assigned to backup device.
                         (If bit set, and FIBHDW = 04, means
                         PBT only; bit reset and 42 moved to
                         FIBHDW.)
                    :8   Specifies that program is controlling
                         descriptor addresses or by the @CF@
                         character (MT7); MCP does not affect
                         FIB-BB or use full MTP I/O error
                         recovery facilities.
FIBRRN     2-  6 5 N   Rerun Number
                    CRD, PCH, PRN, MTP, PTR, SEQ DSK:
                    Number of records left to process before
                    next breakout (checkpoint); decremented
                    to zero, then reset from FIBRRC.
FIBRRC     7- 11 5 N   Rerun Control
                    CRD, PCH, PRN, MTP, PTR, SEQ DSK:
                    Number of records to process between
                    breakouts; moved to FIBRRN at each
                    breakout.
```

FILE INFORMATION BLOCK FIB===
(Continued)

FIB-BA	12 1 N	Buffering Technique :1 Buffer(s) only, program works directly in buffer. :1/ Buffer(s) and work area, program accesses record work area. Value affects meaning of FIBARB, FIB-NB, FIB-WA. :2 Restart running. :4 DPK: File must reside on a single pack; otherwise, areas are spread over available packs (as constrained by FIBDTK). :8 MTP: During automatic reel swap, buffers currently queued for I/O are to be placed on next reel to insure that file is not written off end of current reel (occurs only if program has no label USE routine).
FIBLBL	13 1 N	File Label Convention Non-DSK output: Specifies type of label to create. 0 = Standard Burroughs label. 1 = Label omitted. 2 = USASII standard label. 4 = Label as per Installation Label Card specification. 8 = MTP output only - use first scratch tape available and maintain same label type. Type inserted into FIBLBL.
FIBALT	14 1 N	Number of Alternate Buffers Total # of buffers -1 (only one buffer used for PCR, PCD, PBD, DCM, PBTB regardless of number declared).

FILE INFORMATION BLOCK FIB===
(Continued)

FIBSTA 15 1 N File Status

Used to establish validity of I/O requests or special handling at OPEN, CLOSE, and EOJ.

0 = All files:
File is OPEN.

1 = All files:
File never OPENed. This value is generally found only before the first OPEN of the file.

2 = All input files except RND, DCM, SOR:
File access restricted. I/O requests are restricted to file CLOSE because EOF has been sensed.

3 = All files:
File CLOSEd. Program has OPENed, then CLOSEd file; file can still be attached to device if request was simple CLOSE. See FIBIOA.

4 = CRD:
File prematurely CLOSEd. The MCP has CLOSEd and RELEASEd the unit before the program CLOSE request because a control card has been sensed.

5 = MTP, PTR:
Automatic reel swap in process. When end of reel is sensed, the current reel is CLOSEd and the next reel OPENed.

6 = Premature CLOSE and EOF label taken.

7 = MTP, PTR:
Reel swap in process. When program requests CLOSE REEL, current reel is CLOSEd and next reel OPENed.

9 = MTP:
Multifile search in process. MCP in process of looking for requested file on multifile tape.

FIBSVF 16- 18 3 N Save Factor

MTP:
Number of days beyond creation date that file can be purged automatically by MCP.

```
          FILE INFORMATION BLOCK                        FIB===
                                                           (Continued)

     FIBMRL    19- 23 5 N      Maximum Record Length
                              All files:
                              Record size in digits.
                              For variable length records FIBMRL gives
                              maximum record size allowed.
                              The MCP forbids a value of zero and, for
                              work area access files, a value of 40,000
                              or greater.
     FIBRPB    24- 26 3 N      Records per Block
                              DSK, MTP:
                              Number of fixed length records per block;
                              used to create or recalculate DFH for disk
                              files during MTP/DSK positioning and
                              RND READ/WRITE.
     FIBARB    27- 32 6 N      Addresses of Record in Buffer
                              All files:
                              Meaning depends on FIB-BA. If FIB-BA:1 set,
                              after READ, FIBARB points to record
                              requested; prior to WRITE points to
                              available space in buffer (buffer is
                              available at this point). If FIB-BA:1 not
                              set, prior to READ points to record to be
                              requested; prior to WRITE points to space
                              for next output record (buffer may not
                              be available at this point).
                              If FIBIX2:1, value placed in program
                              IX2 after each I/O request.
     FIBSPF        33 1 N      Flags for Special Forms, Miscellaneous
                              :1  PRN, PCH:
                                  Special forms required.  Automatic
                                  assignment of output device (except
                                  forced backup) cannot be done because
                                  of need for special forms; PRN, PCH
                                  device saved (SV) automatically when
                                  device released.  If backup file
                                  created, flag carried to file.
                              :2  DSK:
                                  Directs MCP to insert the pass file
                                  number into the second and third
                                  characters of the file ID.  This
                                  number is maintained within the MCP
                                  and is incremented at each use.  Used
                                  to ensure file name uniqueness when
                                  <mix no> is not satisfactory.
                              :4  PRN: auto print flag
                              :8  Available
```

FILE INFORMATION BLOCK FIB===
(Continued)

```
FIB-WA    34- 39 6 N      Work Area Address
                        All files:
                        If FIB-BA:1/, address of work area.
                        If FIB-BA:1, unused.
FIBHDW    40- 41 2 N      Hardware Type
                        All files:
                        Prior to file OPEN, specifies hardware type
                        requested (See FIBST1:4 and :8;
                        FIBST2 :2 and :4 for supplementary
                        specifications).  Can be modified in file
                        OPEN due to file equate or IL.  Refer to
                        Section 2 for the specific values.
FIB-IO        42 1 N      Input/Output Mode Flag
                        Establishes validity of current I/O
                        request.
                        0 = File OPENed input.
                        1 = File OPENed output.
                        2 = File OPENed I/O (DSK, DPK, DCM only).
                        3 = File OPENed O/I (RND only); changed
                            to two during file OPEN.
                        4 = Extend/Input (during EOF search).
                        5 = Extend/Output (Note that 4 is
                            converted to 5).
FIBMOD        43 1 N      Recording of File
                        Applies to output files only.  Input file
                        recording mode determined from file
                        (MT7 or label (CRD); For input MTP, parity
                        reflected in FIBMOD.
                        0 = Non-standard recording.
                            PCH:  File in BCL.
                            MT7:  File to be written in even
                                  parity.
                        1 = BINARY recording.
                            PCH:  File to be punched BINARY (low
                            order six bits of even bytes to be
                            punched in top rows of card, low order
                            six bits of odd bytes in bottom rows);
                            file cannot go to backup.
                            MT7: File to be written odd parity.
                        2 = Standard recording.
                            PCH: File to be written in EDCDIC.
```

```
          FILE INFORMATION BLOCK                          FIB===
                                                                (Continued)

   FIBBLK         44 1 N      Blocking Technique
                            0 = All files:
                                Records unblocked.
                            1 = All files:
                                Fixed number of fixed length records
                            2 = All files except DSK, DCM, SOR:
                                Variable number of variable length
                                records per block. For work area access
                                files, output blocks are packed to
                                maximum possible (written only when
                                current record will not fit in space
                                remaining in buffer).  For buffer
                                access files, output blocks are written
                                when the maximum record size (FIBMRL)
                                exceeds the remaining space and are
                                less efficient.
   FIBFNM    45- 46 2 N       File Number
                            File number assigned to file by compiler;
                            value assigned by order in which files
                            declared; used in file equating as index
                            to proper file equate block on disk.
   FIBLBA    47- 48 2 N       Last Buffer to Access
                            Non-DCM:
                            FIB relative index (trailing zero assumed)
                            to last buffer status block; used in
                            buffer rotation.  See FIB-NB.
   FIBCBS    49- 54 6 N       Current Buffer Size
                            All files except RND, SOR, DCM:
                            Size in digits of remaining space in
                            buffer when record length (FIBMRL for
                            fixed  length records or current record
                            size for variable length output) exceeds
                            FIBCBS, physical I/O is triggered (for
                            input files, FIBARB is also compared to
                            FIBACE).
   FIBRCT    55- 62 8 N       Record Count (redefines FIBACT)
                            Non-RND:
                            Number of unique data records accessed
                            (not access to record) during file
                            processing; placed in output MTP ending
                            labels; EOF pointer for output sequential
                            DSK files.
  +FIBACT    55- 62 8 N       Actual Key Storage for RND (redefines
                            FIBRCT)
                            RND:
                            Used as work area to hold the current
                            reccrd number (actual key).
          FILE INFORMATION BLOCK                       FIB===
                                                            (Continued)

FIBMBS   63- 69 6 N      Maximum Block Size
                        All files except SOR, DCM:
                        Buffer size (block size) in digits; gives
                        actual size of blocks of fixed length
                        records (except final block of MTP reel
                        which may be short). Gives maximum block
                        size for variable length records.  Moved
                        to FIBCBS at physical I/O.
FIB-NB   69- 71 3 N      Next Buffer Pointer
                        All files:
                        FIB relative index to current BSB; meaning
                        depends on FIB-BA.
                        If FIB-BA:1 set:  after READ and prior to
                        WRITE points to BSB for current buffer
                        (buffer is available).
                        If FIB-BA:1 not set: prior to READ and
                        WRITE points to BSB for next request
                        (buffer may not be available).
                        Buffer rotation involves modifying FIB-NB
                        (incrementing by 40 or resetting to 200
                        when value exceeds FIBLBA x 10.).
FIBIOA   72- 77 6 N      Input/Output Assignment Table Index
                        All files:
                        Address of IOAT (physical file table)
                        entry for device attached to FIB; file
                        can be OPEN or CLOSEd and not released;
                        contains zeros if no file attached.
          FILE INFORMATION BLOCK                          FIB===
                                                            (Continued)

     FIBTRN       78  1 N      Hardware or MCP Controlled Translation Type
                              MTP, PTP, PTR, DCM:
                              Type of code translation to be done on
                              data before releasing buffer to program;
                              for MTP, PTR, PTP, moved to FIB-D3 except
                              as noted.
                              0 = PTR, PTP:  Process 7-bit odd parity
                                  (append or strip high order bit).

                                  All files except PTP, PTR: No
                                  translation.

                              1 = PTR, PTP:  Translate BCL/EBCDIC (6-bit
                                  odd parity).
                                  MT9: Translate ASCII/EBCDIC
                              2 = PTR, PTP:  Process 8-bit, no parity
                                  (no translation).

                                  DCM: (input):  Non-standard
                                  translation, codes which have both
                                  upper and lower case characters, such
                                  as PTTC/6 and ASCII, have the lower
                                  case set translated to upper case
                                  EBCDIC.
                              4 = DCM: Standard MCP translation.
                                  MT7:  Enable hardware translation for
                                  BCL/EBCDIC conversion.
                                  MT9: Translate ASCII/EBCDIC
                              5 = MT9:  Enable hardware ASCII/EBCDIC
                                  translation.
     FIBOPT        79 1 N      Optional File Flag and Miscellaneous
                              Input files:
                              :1  File declared OPTIONAL: need not be
                                  present (end of file is forced on
                                  first access if OF message entered).
                              :2  MTP:
                                  MCP should display file label
                                  information on SPO at file OPEN.
                              :4  MTP (output):
                                  1600 BPI tape (9-track) permitted
                                  for file.  Bit is set at file OPEN if
                                  FIBST1:8 already set.  At end of file
                                  OPEN, bit left on if MPE assigned
                                  (input or output).
                              :8  MTP (output):
                                  250 IPS tape (GCR) permitted for file.
          FILE INFORMATION BLOCK                          FIB===
                                                              (Continued)

FIBLRA    80- 83 4 N     Logical Records per Area
                        DSK (output, O/I only):
                        Maximum number of records per disk area;
                        MCP uses field only if FIBRPA is zero.
FIBNAR    84- 85 2 N     Number of Areas
                        DSK (output, O/I only):
                        Maximum number of areas for disk file;
                        ignored on input and I/O files (value in
                        DFH is used).
FIB-DA        86 1 N     Disk Access Technique
                        DSK:
                        :1  Random.
                        :1/ Sequential.
                        :2  Sequential I/O.
                        :4  Usercode flag for LOADMP and
                            PACKUP.
                        :8  File declared SHARED (HPT only)
                            indicates that FIBMRL and FIBRPB must
                            match size in DFH at file OPEN;
                            allows shared disk requests when SHRD
                            option set.
FIBDFN    87- 88 2 N     Disk File Number
                        DSK:
                        File Number of disk file (independent of
                        FIBFNM) assigned sequentially according to
                        order of declaration of disk files in
                        source program; used in HPT space
                        assignment if FIBDTK:1.
FIBRSW    89- 93 5 N     Record Size in Words
                        Size of record in words for fixed length,
                        work area access records (FIBMRL/4); (size
                        for variable length calculated from size
                        field in each record); cannot be zero or
                        greater than 9999.
FIBEXT    94- 99 6 N     Pointer to Security Attribute Storage Area
```

FILE INFORMATION BLOCK FIB===
(Continued)

FIBRAD 100 1 N Read Address Flag; Block Access Flags
Input files:
Bits :1 and :2 control the method of determining the ending block address and the direction in which the file is accessed.
If neither bit is set, the file is implied to be read forward; fixed length records and the ending address are taken from FIB-BB since the hardware RAD is not needed (that is, unblocked file or DSK).
If the bits are set, RAD is used.
:1 MTP: Read file forward.
:2 MTP, SEQ DSK: Read file reverse.
The following bits apply to DSK files:
:4 DSK:
After WRITE, perform READ CHECK operation for parity check.
:8 RND:
Do not examine buffers for block; physical read must be performed (READ and blocked WRITE); explicit SEEK request are ignored. Used to force physical I/O, for example, when programs modify data in buffers and require a fresh copy of the record.

FIBOPN 101 1 N Internal File OPEN Flag
All files:
Miscellaneous flags used internally during file OPEN.
:1 Initial file OPEN.
:2 HPT output: file is output pseudo deck (ID=#00000).
:4 File is output breakout file.
:8 Reserved.

FIBADR 102-109 8 N Relative Disk Address of Block in area (redefines FIBSQ1, FIBSQ2, FIBSQ3, FIBSQL, FIBQAD).
RND:
Used as work area to build the relative disk address (within the area) for the current operation.

FIBQAD 102-107 6 N Address of SOR Queue Element (redefines FIBADR, FIBSQ1, FIBSQ2, FIBSQ3, FIBSQL)
SOR:
Absolute address of IOQ element reserved at file OPEN for SOR.

FIBSQ1 102 1 N Previous I/O Request Type (redefines

FILE INFORMATION BLOCK FIB===
(Continued)

```
                                FIBADR, FIBQAD)
                                SEQ I/O DSK:
                                Storage of value of FIBRWT for previous I/O
                                request; set from FIBSQ2 at each request.
                                0 = Request was READ; if current request
                                    also a READ, must change pointers to
                                    access proper record.
                                1 = Request was WRITE; pointers are
                                    correct for current request.
FIBSQ2       103 1 N              Current I/O Request Type (redefines
                                FIBADR, FIBQAD)
                                Storage of value of FIBRWT for current
                                I/O request.
                                0 = Request is READ; if previous request
                                    was also a READ, pointers (FIBARB,
                                    FIBCBS) still point at previous
                                    request and must be changed; otherwise,
                                    pointers are correct.
                                1 = Request is WRITE; must set FIBSQ3:2
                                    to force physical I/O when block is
                                    completed; advance pointers to next
                                    record.
FIBSQ3       104 1 N              Current I/O Request Type (redefines
                                FIBADR, FIBQAD)
                                SEQ I/O DSK:
                                :1  Internal flag specifying both FIBSQ1
                                   and FIBSQ1 indicates READ and must
                                   cycle back through READ/WRITE to
                                   access correct record; bit reset
                                   during return cycle.
                                :2  Specifies write activity on current
                                   block and when the block is exhausted,
                                   must be written.
                                   MTP:
                                :4  Retry short/long records.
                                :8  Ignore short/long records.
FIBSQL   105-109 5 N              Sequential I/O Record Length (redefines
                                FIBQAD, FIBADR)
                                SEQ I/O DSK:
                                Storage area for record length of last
                                record accessed.
FIBORG       110 1 N              File Type
                                DSK:
                                :1 Relative I/O data file.
                                :2 Indexed I/O data file.
                                :4 Indexed I/O key file.
                                :8 Reserved
FIBAVL       111 1 N              Open Result Status on Open Available
```

```
           FILE INFORMATION BLOCK                         FIB===
                                                       (Continued)

                              0 = File not available (Locked, etc.)
                              1 = File available - successful open
                              2 = File non-existant
                              6 = No resources for open (no mem)
                              7 = Pack not available
                              9 = File not created (dup LIB on output)
     FIBDKB  112-119 8 N        Relative Block Number (redefines FIBDCF)
                              RND:
                              Temporary internal storage for relative
                              block number of file during READ/WRITE
                              requests.
     FIBDCF      112 1 N        DCM FILL Flag (redefines FIBDKB)
                              DCM:
                              0 = No FILL given.
                              1 = FILL initiated.
     FIBKEY  120-125 6 N        Actual Key Location (redefines FIBABS,
                                FIBJAM)
                              RND:
                              Address of actual key (8 UN).
                              MTP:
                              Used internally to store first six digits
                              of I/O descriptor during tape position.
     FIBJAM  120-125 6 N        Address of Jam/Missort USE routine
                              (redefines FIBABS, FIBKEY)
                              SOR:
                              Address of USE routine for handling
                              jam/missort/EOF (SORTER 5).
     FIBSBL      126 1 N        Buffer Flag
                              :1  New area OPENed by GETDSK.
                              :2  DPK:
                                  File is DPK file.
                              :4  HPT:
                                  File is HPT or backup disk.
                              :8  All files except RND, DCM, SOR:
                                  Buffer required flag; buffer exhausted
                                  on previous I/O request, new buffer
                                  needed for further processing (used for
                                  all work area access files except
                                  variable length).
     FIBUNF      127 1 N        Random Disk Wait Flag; Tape Position Flag
                              RND:
                              Specifies program waiting for RND
                              processing.
                              1 = Buffer required for current request
                                  (I/O not yet initiated).
                              2 = Program waiting I/O on current request
                                  (READ or implicit SEEK for blocked
                                  WRITE).
```

FILE INFORMATION BLOCK FIB===
(Continued)

```
                              3 = Program waiting buffer availability
                                  for next I/O (WRITE in process on
                                  buffer access file).
                                  MTP:
                                  Specifies program waiting for MTP
                                  position.
                              2 = Waiting for all in process I/Os to
                                  complete before positioning begins.
                              4 = Waiting for I/O on SPACE operation
                                  when moving tape.
                              6 = Input file buffer fill in process
                                  after positioning complete.
FIBFLM     128    1 N
                              :1  File limits specified  (DISK).
                              :2  Search for reserved scratch tape.
                              :4  Ignore channel 12 on printer (RPG).
                              :8  DMS-II file (MCP only flag).
FIBIX2     129    1 N           IX2 Flag:  Breakout Flags
                              All files:
                              :1  Directs MCP to move FIBARB to program
                                  IX2 after each READ or WRITE request
                                  (see FIBARB); generally used for
                                  buffer access files.
                              :2  Save previous breakout disk file.
                              :4  Use MTP for breakout (rerun every n
                                  records).
                              :8  Use DSK for breakout (ignored if :4
                                  set).
FIBBCT  130-137   8 N           Block Count
                              All files:
                              Number of unique blocks of data read or
                              written during file processing (value for
                              RLOG taken from IOAT).
                              For MTP, value inserted into output ending
                              labels and checked when tape used as input.
FIBPOS  138-142   5 N           File Position data (redefines FIBNAU,
                              FIBBFF, FIBFFL, FIBDCO, FIBWTF, FIBBCF)
                              PRN, SEQ DSK, MTP:
                              Field used as 4 SN to hold data from POSN
                              communicate.
                              PRN:  Channel or line skipping data (sign
                              digit ignored).
                              MTP,DSK:  Number of records to skip (sign
                              digit gives direction).
                              DISK:@F@ in last digit causes EOF action
                              to be taken on the next disk request; The
                              second digit is used as an internal flag
                              with @F@ meaning all buffers refilled.
```

```
          FILE INFORMATION BLOCK                          FIB===
                                                       (Continued)

   FIBPSN  138-142 5 N      Redefine FIBPOS
                          TAPE:
                          FIBPOS as 4SN.
   FIBNAU  138-140 3 N      Number of Disk Areas (redefines FIBPOS,
                          FIBBFF, FIBFFL).
                          DSK (output only):
                          Number of disk areas assigned during run.
                          Used internally during CLOSE.
   FIBDCO      140 1 N      Stream Mode OP Storage (redefines FIBPOS,
                          FIBNAU)
                          Holds low order digit of DCM OP code for
                          current request.
   FIBWTF      141 1 N      Wait Flag (redefines FIBPOS)
                          DCM (stream mode):
                          1 = Waiting IOC.
   FIBUSE  143-144 2 N      USE Routine Exit Handling
                          Used as branch table key when USE routine
                          exited; identifies type of routing in
                          process.
                          00 = No further MCP action needed (input
                               begin label); stalemate (shared disk).
                          06 = Output begin label (label still to be
                               written).
                          12 = Output end label (label still to be
                               written).
                          18 = Input end label (CLOSE to be
                               completed).
                          24 = End-of-page routine (no further
                               action needed).
                          30 = WRITE parity routine (logical I/O to
                               be completed if work area access).
                          36 = MCS DCP - write error.
                          42 = READ parity routine.
   FIBRWT     145  1 N      READ/WRITE Type
                          All files except SOR, DCM:
                          Type of I/O request tc begin process. Bits
                          1 and 2 specify the general type of request
                          as follows.
                          0      READ request.
                          1      WRITE request.
                          2      RND:  SEEK request.
                          3      PRN, MTP, SEQ DSK:  Position request.
                          RND:
                          In addition to the above, bits 4 and 8
                          specify shared disk operations yielding
                          the following possible digit values:

                          0 = Plain READ (ignore lock status).
          FILE INFORMATION BLOCK                                  FIB===
                                                             (Continued)

                                  1 = Plain WRITE (unlock implied if
                                      relevant).
                                  2 = Plain SEEK (ignore lock status).
                                  3 = Not valid for RND.
                                  4 = LOCK (wait if currently locked).
                                  5 = WRITE NO UNLOCK (block must have been
                                      locked previously by requestor).
                                  6 = LOCK SEEK (do not wait if currently
                                      locked).
                                  7 = Not valid.
                                  8 = READ WITH LOCK (wait if currently
                                      locked).
                                  9 = UNLOCK (block must have been locked
                                      previously by requestor).
                                  A = SEEK WITH LOCK (do not wait if locked
                                      currently).
                                  B = Not valid.
                                  C = READ UNTIL UNLOCKED (plain READ, wait
                                      if currently locked).
                                  D = Not valid.
                                  E = SEEK UNTIL UNLOCKED (plain SEEK,
                                      retry if currently locked).
                                  F = Not valid.
FIBWKF      146   1 N               Work File Flag; Miscellaneous
                                  :1  DSK:
                                      Insert requestor's mix number in second
                                      and third characters of file ID.  Used
                                      to ensure file ID uniqueness, primarily
                                      for work files.
                                  :2  DSK:
                                      Insert processor number on which
                                      requestor running into fifth character
                                      of file ID.  Used to ensure file ID
                                      uniqueness in shared disk systems.
                                  :4  Call TERM return at end of CLOSE.
                                  :8  Call TERM at end of CLOSE.
FIBDTK      147   1 N               Disk Assignment Technique
                                  DSK (output, O/I):
                                  Directs MCP to assign any disk needed
                                  during execution of program in particular
                                  ways.  Used to improve I/O overlap on
                                  multi-channel or multi-subsystem
                                  configurations and/or to assign files
                                  to particular subsystems.
                                  0 = Use random disk assignment method.
                                      Assign each area (page) to a randomly
                                      selected EU by taking last two digits
                                      from interval timer, dividing by total
```

FILE INFORMATION BLOCK FIB===
(Continued)

```
                                number of EUs on subsystem(s) and using
                                remainder to select an EU of those
                                on subsystem(s). (Often good method
                                for random files.)
                            1 = Use FIBDFN as seed number for method
                                above rather than random number. This
                                method attempts to place entire file on
                                same EU. (Often good method for
                                programs with multiple output
                                sequential files.)
                            2 = Use area number for seed number. This
                                method attempts to place each area of
                                a file on a different EU. (Often good
                                method for random files.)
                            4 = Select EU from value in FIB-EU. If
                                EU not on system use value as seed
                                number. Method attempts to place
                                all new areas on same EU. (Often good
                                method for either random or sequential
                                files when mix is well defined.)
                            :8  Unused.
                            DPK (output), O/I, I/O:
                            Controls the area assignment of DPK files.

                            :1  Assign areas beginning at a cylinder
                                boundary; otherwise space is assigned
                                from the beginning of an available
                                area.
                            :2  Assign each area to successive packs
                                (multipack files only); otherwise,
                                assign space as available (for
                                multipack files, preference given to
                                the pack with the largest amount of
                                available space).
                            :4  Unused.
                            :8  Unused.
FIBCHN  148-149 2 N           Channel Number (redefines FIB-EU)
                            Non-DSK:
                            Primary channel for I/O on this file.
FIB-EU  148-149 2 N           Selected EU for HPT Assignment (redefines
                            FIBCHN)
                            DSK:
                            Used for subsystem and disk space
                            assignment if FIBDTK:4.
                            At file OPEN (output, O/I)
                            If first digit is nine, second digit gives
                            relative number of subsystem desired for
                            file (value must be three or less; zero
          FILE INFORMATION BLOCK                          FIB===
                                                              (Continued)

                                   means default). Indicates all areas of
                                   the file are to be allocated on the
                                   designated subsystem(s). (Subsystem
                                   designation remains with file in DFH.)  If
                                   first digit is not nine, entire field
                                   designates specific EU (by number) for
                                   file.  If EU on system, associated
                                   subsystem is selected; else default
                                   subsystem(s) are used.
      FIBUNT       150 1 N           Unit Number
                                     Non-DSK:
                                     Unit Number of device assigned.
      FIBAUD       151 1 N           Reserved
      FIBULB   152-157 6 N           Label USE Routine Address (redefines
                                     FIBUST, FIBURE)
                                   All labeled files:
                                   Address of USE routine for user label
                                   handling.  Values put in BASE:+34 through
                                   BASE:+39 by MCP, can be used to evaluate
                                   file status:
                                   BASE:+34:1 = FIB-IO (distinguish
                                                input/output).
                                   BASE:+35:1 = 0 for begin label, 1 for
                                                end label.
                                   BASE:+36:1 = 0 for file labels, 1 for
                                                reel labels (MTP).
                                   BASE:+37:3 = Reel number (MTP).
      FIBUST   152-157 6 N           RND DSK:
                                   Address of USE routine for shared disk
                                   stalemate handling.
      FIBURE   152-157 6 N           Read Error USE Routine (redefines FIBUST,
                                   FIBULB)
                                   SOR:
                                   Address of general read error USE routine.
                                   (unencoded, cannot read, and so on,
                                   SORTER 1).
                                   S4A, S4B:
                                   Address of result status attention USE
                                   routine.
      FIBUER   158-163 6 N           I/O Error USE Routine Address (redefines
                                   FIBUAE)
                                   MTP, DSK, PRN, PTP, PTR:
                                   Address of USE routine for I/O error
                                   handling.  Values put in BASE:+34 through
                                   BASE:+39 by MCP, can be used to evaluate
                                   reason for entry:
                                   BASE:+34:1 = FIBRWT bits :1 and :2
                                                (distinguish READ/WRITE
```

FILE INFORMATION BLOCK FIB===
(Continued)

```
                                errors).
                              BASE:+37:3 = Reel number (MTP).
FIBUAE  158-163 6 N             Amount Error USE Routine (redefines FIBUER)
                              SOR:
                              Address of USE routine for amount error
                              handling (SORTER 2).
                              S4A, S4B:
                              Address of soft R/D USE routine.
FIBUEP  164-169 6 N             End-of-page USE routine Address (redefines
                              FIBUTE, FIBPIN)
                              PRN:
                              Address of USE routine entered when
                              12-punch sensed in printer carriage tape;
                              meaningless if file diverted to backup.
                              MTP:
                              Temporary storage for number of blocks
                              to space during positioning.
                              DSK:
                              Address of file limit table (COBOL files
                              having file limits).
                              SEQ DSK (input):
                              EOF pointer value (BSN) from DFH (obtained
                              at OPEN) used by COBOL DEBLOCK routine
                              (also redefines FIBUPS).
FIBUTE  164-169 6 N             Address of Transit Error USE Routine
                              (redefines FIBUEP, FIBPIN)
                              SOR:
                              Address of USE routine to hanole transit
                              field errors (SORTER 3).
FIBPIN  164-169 6 N             Ping Address for DCM Stream Mode
                              (redefines FIBUEP, FIBUTE)
                              DCM (stream mode):
                              Address of beginning of buffer area.
FIBEOF  167-175 8 S             Disk File EOF Pointer
                                (redefines FIBUEP, FIBUTE, FIBPIN, FIBUPS)
                              DSK:
                              Disk File EOF pointer.
FIBUPS  170-175 6 N             Pocket Select USE Routine Address
                              (redefines FIBPON)
                              SOR, S4A, S4B:
                              Address of pocket select USE routine
                              (no error) (SORTER 4)
                              HPT:
                              Current file limit table entry address
                              (COBOL programs having file limits).
FIBPON  170-175 6 N             Pong Address for DCM Stream Mode
                              (redefines FIBUPS)
                              DCM (stream mode only):
```

FILE INFORMATION BLOCK FIB===
(Continued)

Address of second half of buffer area (put in FIB-BB).

FIBRPA 176-183 8 N Number of Records per Disk Area

DSK (output, O/I):
Number of records declared per disk area.

DSK (input, I/O):
Value calculated from values in DFH and FIBRPB; used only if FIB specifies FIBMRL and FIBRPB different from values in DFH.

MTP:
Temporary storage for number of blocks to move tape during positioning.

FIBCOD 184-185 2 N Descriptor OP Code Storage

All files except RND, DCM, SOR, SEQ I/O:
Contains OP code for processing of file; placed into FIB-OP during READ/WRITE; OP code for other file types developed at READ/WRITE time.

DCM:
Temporary storage for request type from communicate.

FIBBC1 186 1 N OPEN/CLOSE Communicate Variant

All files:
Holds first variant digit from OPEN/CLOSE communicate during OPEN and CLOSE. See FIBBC2.

FIBBC2 187 1 N OPEN/CLOSE Communicate Variant

All files:
Holds second variant digit from OPEN/CLOSE communicate during OPEN and CLOSE. See FIBBC1.

FIBLAB 188-193 6 N All files:

Addresses the byte before the MFID field of the label area. The size and format of the area is dependent on the file and label type definitions. All files have at least a minimum label area containing the file name fields. Addresses must be modulo 4.

FIBLAE 194-199 6 N All files:

This field addresses the 1st digit past the label area. Depending on the file and label type definitions, it may function as the end address in I/O descriptors for label operations. Addresses must be modulo 2 or 4 depending on hardware and label type declared.

FILE BUFFER DESCRIPTORS FIB===

File Buffer Descriptors (also called Buffer Status Blocks) are located immediately following the FIB for a file. There is one Buffer Status Block (BSB) for each buffer declared for the file. (Note that three alternate areas means four buffers.)

Each BSB is 40 digits long. Each is set up by the compiler and used to handle the I/O operation that is to be performed on that buffer.

A BSB can be addressed relative to the FIB. For example, if a file has three buffers, there is a BSB at FIB:+200, FIB:+240, and FIB:+280.

The following descriptions are BSB-relative instead of FIB-relative.

FIBBSW 0- 3 4 N Buffer Status Word

All files:
The first three digits of this field contain the high order digits of the hardware result descriptor (R/D) which occurred at the completion of the physical I/O operation for this buffer. If the R/D indicates an error condition, the last digit contains a code describing the particular error.

=8000	No exception
=9012	Write or unlock to record not locked by requestor (shared disk)
=9022	I/O timeout (no R/D returned)
=9102	End address limit error
=9202	Attempt to write MCP disk
=9402	Invalid I/O (hardware detected)
=???1	Invalid R/D
=???2	Invalid I/O Descriptor (see table 5-2)
=???3	Parity error
=???4	Memory parity error during I/O
=???5	Not ready
=???6	End of file (for example, tape mark)
=???7	End of medium (for example, EOT)
=???8	Interface Parity Error
=???9	Reserved
=???A	Short record (MTP)
=???B	Long record (MTP)
=???C	End-of-page (PRN)
=???D	Reserved
=???E	Special error ignored

```
          FILE BUFFER DESCRIPTORS                         FIB===
                                                            (Continued)

                                  =???F      All errors ignored

                                  Where the three question marks are the  first
                                  three  digits of the (converted) exception RD
                                  (possible values C00-FFF, since both 8 and  4
                                  bit of first digit are on for exceptions).

     FIB-OP     4-   5 2 N          I/O Descriptor Code
                                  All files:
                                  OP code for descriptor.  Set from FIBCOD
                                  for files other than RND, SEQ I/O DSK,
                                  SOR and DCM.  For these files, value
                                  developed during READ/WRITE.  Contains
                                  @FF@ if file OPENed by optional file
                                  mechanism.
     FIB-D1          6 1 N          Variant Digit for Descriptor
                                  Value depends on file type; not used for
                                  some types.  Some uses are:

                                      Part of HPT EU specification.
                                      MTP density specification.
                                      PRN/PCH spacing/stacker select.
     FIB-D2          7 1 N          Variant Digit for Descriptor
                                  DSK, MTP, PRN:
                                  Generally contains unit number designation
                                  of device attached.  Meaningless for some
                                  hardware types.
     FIB-D3          8 1 N          Variant Digit for Descriptor
                                    Meaning varies with hardware type.
     FIB-D4          9 1 N          Variant Digit for Descriptor
                                    Meaning varies with hardware type.
     FIB-AA    10-  15 6 N          Beginning Address of Buffer
                                  Address of beginning of buffer.
                                  Value must be modulo 4.  Acts as
                                  terminating address for read backward
                                  operations (MTP).
     FIB-BB    16-  21 6 N          Terminating Address of Buffer
                                  Address of digit following buffer.  For
                                  stream mode DCM, address of second half
                                  of buffer.  Value must be even, and
                                  must be modulo 4 for MTP, DSK.  Acts as
                                  terminating address for all input
                                  operations and all output operations except
                                  PRN, PCH.
     FIB-AD    22-  27 6 N          Disk File Address
                                  DSK:
                                  Low order six digits of disk address
                                  requested on READ or WRITE operation
```

FILE BUFFER DESCRIPTORS FIB===
(Continued)

(high order digit consists of low order two bits of FIB-D2.
Temporary storage for enable R/D (low order four digits of field).

FIBRCW 28- 33 6 N Return Control Word (redefines FIBBL#, FIBFSA).
Files having USE routines except SOR:
Program address to which return must be made after USE routine exited. (FIBRCW used in first BSB only.)

FIBBL# 28- 35 8 N Random Block Number (redefines FIBRCW, FIBACE, FIBFSA, FIBBBA).
RND:
Zero relative block number of block currently in buffer; used to oetermine if physical I/O needed to access desired record (if FIBRAD:8 not set).

FIBFSA 28- 33 6 N Flow Stopped Address (redefines FIBBL#, FIBRCW)
SOR (first BSB only):
Address of flow stopped label taken from communicate parameters.
S4A, S4B:
Address of fatal R/D label (descriptor error, invalid BCT, invalid flow conditions, invalid IIO error).

FIBACE 34- 39 6 N Actual Ending Address of Buffer (redefines FIBBBA, FIBBL#, FIBHDK)
All files except SOR, RND:
Ending Address of block in buffer. May differ from FIB-BB for short blocks (for example, final blocks on MTP reel or variable length records). Set from hardware RAD operation or FIB-BB depending on FIBRAD value. Used to detect end of buffer and need for physical I/O on input files. See FIBCBS.

SOR:
Storage for time interval passed if program is to be aborted due to pocket select routine taking too much time.

FIBBBA 34- 39 6 N Black Band Address (redefines FIBACE, FIBBL#, FIBHDK)
SOR (first BSB only):
Address of black band label taken from communicate parameters.
S4A, S4B:

FILE BUFFER DESCRIPTORS FIB===
(Continued)

```
                             Address of flow stopped label
FIBHDK   36- 38 3 N            Hashed Disk Address (redefines FIBACE,
                             FIBBBA)
                             RND:
                             Randomized value of disk address for
                             current buffer (first three digits and
                             last three digits of FIB-AD NORed
                             together); used as check beyond FIBBL#
                             to determine presence of requested block.
FIBSEK       39 1 N            SEEK Buffer Flag (redefines FIBACE,
                             FIBBBA)
                             :8  Buffer used for explicit SEEK.
                             :4  Reserved
                             :2  Reserved
                             :1  Reserved
```

FILE PARAMETER BLOCK FP-===

If a program declares any files, the compiler builds File Parameter Blocks (FPB) following the empty segment after the PPB. These blocks are used only for FILE (label) equating. When a FILE equate control card is entered with a program request, the MCP checks the internal file name in the FPB against the name in the FILE equate card. If a match is found, all FPBs are copied into a separate disk area (if an area has not been obtained for a previous label equation request for the program), and the file equate information (name change, device change) is written into the appropriate record in the copy.

At file OPEN time, the MCP accesses the appropriate record in this label equate block (using FIBFNM as an index) and, if FILE equation was done for the file being OPENed, the necessary modifications are made to the FIB and/or label areas.

FIBs are 100 or 200 digits in size depending on the compiler. PB-FPF in the PPB defines the size. If a program code file contains an odd number of 100-digit FPBs, the last 100 digits at the end of the FPB area are unused.

FILE PARAMETER BLOCK FP-===
(Continued)

FPB on Disk

The format of the FPB on disk is as follows.

FP-FNM	0- 11	6	A	Internal file name
FP-MFD	12- 23	6	A	Multifile-ID
FP-FID	24- 35	6	A	File-ID
FP-HWR	36- 37	2	N	Hardware type
FP-BUP	38	1	N	Backup flag
FP-LEQ	39	1	N	Label Equate flag
FP-SPF	40	1	N	Special Forms flag
FP-TRK	41	1	N	Magnetic tape track type
FP-GRD	42- 53	6	A	Guard file ID
FP-STY	54	1	N	Security type (refer to SASA)
FP-FIB	55- 60	6	N	Base-relative FIB address
FP-SEG	61- 63	3	N	Number of logical segments containing this FIB
FP-SUS	64	1	N	Security use (refer to SASA)
	65- 76	12	N	Reserved
FP-SNS	77	1	N	Sensitive data flag
FP-FAM	78- 89	6	A	Pack ID of guard file (default = DISK)
FP-MA1	90	1	N	Security Maint (not implemented)
	91	1	N	Reserved
FP-HST	92-125	17	A	Hostname
FP-RPA	126-133	8	N	Disk/Pack records per area
FR-#AR	134-137	4	N	Disk/Pack number of areas
	138-199	62	N	Reserved

MCS BUFFER DESCRIPTION IH-===

There is one MCS Buffer for each and every MCS running in the system. It contains a soft IOAT (Pseudo IOAT), the input header pool, pointers for the input header pool for this MCS, the external buffer, and pointers to the external buffer and a table that contains the LSM of the lowest numbered not ready station on each DCP that is attached to this MCS. The specific information contained in the MCS buffer is shown on a memory dump under the following headings:

1. DCP buffer analysis (IH-=== in MCP listing).

2. User External Buffer DCP Header which contains the block of information to be passed to the MCS on its next read of the DCP file, or the next block to be passed to a DCP.

This MCS Buffer is contained in a Type 4:Subtype 1 memory block, with the mix number of the MCS assigned to it.

INPUT HEADER QUEUE BUFFER - DCP

MCS/USER INPUT HEADER QUEUE, EXTERNAL BUFFER

IH-DAT	0- 99	99	N	SOFT IOAT (MCS FILE)
IH-RCS	100-103	4	N	MCS RECORD SIZE LESS HEADER (BYTES)
IH-REC	104-111	8	N	MCS FILE RELATIVE RECORD ADDRESS
IH-DUM	112-119	8	N	MCS TABLE ENTRY ADDRESS (Absolute)
IH-1AV	120-127	8	N	FIRST AVAIL HEADER SLCT IN HEADER POOL BUFFER RELATIVE
IH-HDR	128-135	8	N	ADDR OF NEXT HDR IN POOL TO BE PROCESSED BUFFER RELATIVE
IH-SLT	136-139	4	N	NUMBER OF HEADERS IN POOL
IH-IOB	140-147	8	N	BFR-REL ADDR OF EXTERNAL I/O BFR
IH-MAX	148-151	4	N	MAX EXTERNAL I/O BFR SIZE LESS HEADER(BYTES)
	152-155	4	N	RESERVED
IH-ID#	156-157	2	N	MCS ID NUMBER
IH-CHN	158-159	2	N	CHANNEL # OF ACTIVE I/O (= FF IF NONE)
IH-LHQ	160-199	40	N	FIRST LSN ON EACH DCP WHICH IS NOT READY (THIS MCS ONLY)
IH-DPH	200-END			80 DIGIT HEADER POOL ENTRIES AND EXTERNAL BUFFER

INSERT FILE IND-==

This file is used for Insert and for asynchronous accept data.

INDISK	0-199	200	N	INSERT FILE RECORD
IND-LN	0- 15	16	N	LINK TO NEXT RECORD
IND-AD	16- 21	6	N	RELATIVE TO PROGRAM ADDRESS
IND-TP	22	1	N	TYPE OF ENTRY (0 = UA INSERT, 1 = UN INSERT, 2 = AX MESSAGE)
IND-LG	23- 24	2	N	LENGTH OF ENTRY
IND-NX	25- 27	3	N	LINK TO NEXT ENTRY IN RECORD (RELATIVE TO FIRST IND-AD)
IND-TX	28-nnn	n	A	TEXT OF ENTRY
	nnn-199			MORE ENTRIES AS ABOVE

INPUT/OUTPUT ASSIGNMENT TABLE IO-===

The Input/Output Assignment Table (IOAT) is also known as the Device Assignment Table (DAT). For each hardware unit declared to the MCP, there is a corresponding entry in the IOAT.

The physical attributes of a hardware device are encoded into an IOAT entry. Some of the information contained in an IOAT entry includes hardware type, channel, unit, device status, disk pack type, and translation type.

When a program OPENs a file, an IOAT entry, corresponding to the harware type requested or allowed to the file, is assigned to the program. This same IOAT entry remains attached to the program until the associated file is CLOSEd RELEASE or the job goes to EOJ.

Most hardware devices on a system can only be accessed by one program, for example, only one program can be using a printer. In such cases, the IOAT entry is linked to the program. Disk type devices and the DCP can be accessed by multiple users. But the MCP only maintains one IOAT entry for each of these units. When a program OPENs a file assigned to one of these hardware types, the MCP makes a copy of the actual IOAT entry and attaches the copied version to the program. All the file actions that the program performs will refer to this copied (soft) IOAT.

For disk type files, the soft IOAT is allocated along with the file header in 200 digits of contiguous memory. For a DCP file, the IOAT is allocated in the program external DCP buffer.

Programmatic access to the IOAT of a file is not possible since the IOAT entry is outside the bounds of the program BASE and LIMIT.

The MCP also maintains an IOAT on disk. This IOAT is a skeleton table which indicates what channels, units, and device type are declared to the system. From the disk version, the in-memory IOAT is built when the MCP is HALT/LOADed.

The following describes the information found in an IOAT entry. The first 68 common digits are described first followed by separate definitions for various device classes for digits 68-99.

```
IO-HDW     0-  1  2 N    Hardware Type
IO-HDS         2  1 N    Supplementary Hardware Type
```

```
                INPUT/OUTPUT ASSIGNMENT TABLE                  IO-===
                                                                 (Continued)

                                    TAPE:
                                    8 = GCR
                                    4 = PE
                                    2 = 9 Track
                                    1 = 7 Track
                                    CARD READER /PUNCH:
                                    0 = Standard 80 COL card reader/punch
                                    PRINTER:
                                    0 = Standard Printer
                                    1 = Train Printer
                                    4 = Translate Table Printer (UIO DLP)
                                    SORTER:
                                    0 = All others
                                    1 = 4A
                                    2 = 4B
                                    SYSTEM SPO
                                    1 = TC 4000
                                    0 = All others
                                    DISK PACK:
                                    0 = Type unknown
                                    1 = 215 Pack
                                    2 = 225 Pack
                                    3 = 235 Pack
                                    4 = 206 Pack (interlaced)
                                    5 = 206 Pack (sequential)
                                    6 = 207 Pack (interlaceo)
                                    7 = 207 Pack (sequential)
          IO-UNT         3   1  N     Unit Number
          IO-CHN    4-   5   2  N     Primary I/O Channel
          IO-LNK    6-   7   2  N     I/O Queue Access Link
          IO-STA    8-   9   2  N     Device Status Digit Link
          IO-QUE   10-  11   2  N     I/O Queue Element Count
          IO-MIX   12-  13   2  N     Mix Number of User
          IO-FIB   14-  19   6  N     User FIB Address (base relative)
          IO-MSK   20-  23   4  N     I/O Error Ignore Mask
          IO-NSC        24   1  N     Unit Status Digit
                                    0 = Unit ready and unassigned
                                    1 = Unit not ready
                                    2 = Control load required (TPR, DPK, DCP)
                                    3 = Waiting, not ready
                                    4 = Reserved
                                    5 = DPK waiting, ready for initial status
                                    6 = MTP waiting, ready for initial status
                                    7 = PTP waiting, ready for rewind
                                    8 = Reserved
                                    9 = Device initialization requested
                                    A = Unit bypass (not tested or assigned)
                                    B = Unit ready (assigned: not to be tested)
```

INPUT/OUTPUT ASSIGNMENT TABLE IO-===
(Continued)

				C = Unit not available: slow/no IOC during status
				D = Unit not available: Invalid I/O during status
				E = Unit requires STAT-2
				F = Unit not available: XU-ed
IO-ERC	25- 27	3	N	Total Error Count on File
IO-ERT	28- 29	2	N	Total Retry Count
IO-BCT	30- 37	8	N	Block Count
IO-ID	38- 49	6	A	File Identifier
IO-MFD	50- 61	6	A	Multifile Identifier (all except DCP)
IO-FFD	50- 61	6	A	Firmware File Identifier (DCP)
IO-USE	62	1	N	Device Usage
				0 = Standard usage
				1 = Direct I/O
				2 = Trace Printer
				3 = Pseudo-reader deck (pack output)
				4 = Pseudo-card reader (pack)
				5 = Punch backup pack
				6 = Printer backup pack
				7 = Pseudo-reader deck (disk output)
				8 = Pseudo-card reader (disk)
				9 = Punch backup disk
				A = Printer backup disk
				B = Printer backup tape blocked
				C = Printer backup tape unblocked
				D = Reserved
				F = Physical file is on remote host
IO-WIO	63	1	N	
				:8 Waiting close queue flush
				:4 Queued I/O required for N-SEC testing
				:2 Waiting position or file flush
				:1 Waiting I/O complete
IO-ST1	64	1	N	
				:8 Inhibit I/O on this file
				:4 Unit in use
				:4/ Unit available
				:2 Input
				:2/ Output
				:1 Open
				:1/ Closed
IO-ST2	65	1	N	
				:8 Call terminate at end of CLOSE
				:4 CLOSE called by terminate
				:2 CLOSE called by exception processing
				:1 EOF sensed
IO-ST3	66	1	N	
				:8 File in use by SORT intrinsic

INPUT/OUTPUT ASSIGNMENT TABLE IO-===
(Continued)

```
                                  :4  Unit is DLP type device
                                  :2  User table entry has been made
                                      (unit logged on)
                                  :1  User program not in memory
     IO-ST4      67  1 N
                                  :8  Time-sharing IOAT
                                  :4  Reserved
                                  :2  TSM IOAT pushed but FIBIOA not updated
                                  :1  TSM process = type I
                                  :1/ TSM process≠ type II
```

INPUT/OUTPUT ASSIGNMENT TABLE IO-===
(Continued)

DISK (also used for soft pack IOATS)

IO-ADR	68- 75	7	S	Memory Address of Next Disk Address
IO-AR#	76- 77	2	N	Current Area Number
IO-FS1	78- 79	2	N	FPM Slots Assigned
IO-EOF	80- 87	8	N	DF-EOF Storage for remote host disk file
*IO-FS2	80- 81	2	N	FPM Slots In Use
IO-RBA	82- 88	7	N	Remaining blocks in area
IO-DSK	89	1	N	OPEN Type
				:8 Random
				:8/ Sequential
				:4 OPENed reverse
				:2 COBOL code file
				:1 Standard code file
IO-DK2	90	1	N	
				:8 Waiting address block memory for file OPEN
				:4 File declared as SHARED (SHRD DISK)
				:2 File had breakout
				:1 Reserved
IO-CLA	91	1	N	File Classification
				:8 Private
				:4 Information
				:2 Public
				:1 Free
				:1/ Control
IO-PK1	92	1	N	
				:8 Base pack not resident on system
				:4 Base pack type restricted
				:2 Base pack type master
				:1 Reserved
IO-PK2	93	1	N	
				:8 Pack overflow specified
				:4 Waiting delayed OPEN (Pack)
				:2 Waiting powered off in use pack
				:1 Reserved
				Read Only File (Unit write locked out at OPEN)
IO-HPT	94- 99	6	N	Disk File Header Address (divided by 10)

INPUT/OUTPUT ASSIGNMENT TABLE IO-===
(Continued)

SPLIT CYLINDER DISK PACK

IO-ADR	68- 75	8	N	Disk Address Next I/O
IO-AR#	76- 77	2	N	Current Area Number
IO-RPA	78- 81	3	N	Remaining Partitions in Area
IO-RBP	82- 88	7	N	Remaining Blocks in Partition
IO-DSK	89	1	N	OPEN Type
				:8 Random
				:8/ Sequential
				:4 OPENed reverse
				:2 COBOL code file
				:1 Standard code file
IO-DK2	90	1	N	
				:8 Waiting address block memory for file OPEN
				:4 File declared as SHARED (SHRD DISK)
				:2 File had breakout
				:1 Reserved
IO-CLA	91	1	N	File Classification
				:8 Private
				:4 Information
				:2 Public
				:1 Free
				:1/ Control
IO-PK1	92	1	N	
				:8 Base pack not resident on system
				:4 Base pack type restricted
				:2 Base pack type master
				:1 Reserved
IO-PK2	93	1	N	
				:8 Pack overflow specified
				:4 Waiting delayed OPEN (Pack)
				:2 Waiting powered off in use pack
				:1 Reserved
IO-HPT	94- 99	6	N	Disk File Header Address

INPUT/OUTPUT ASSIGNMENT TABLE IO-===
(Continued)

STANDARD DEVICES

IO-UST	68- 70	3	N	Result Descriptor Last Status Update
IO-LKS	71	1	N	Unit Status
				:8 Unit Saved
				:4 Unit to be Saved
				:2 Saved by MCP (secure or backup)
				:1 Unit Locked
IO-LBL	72	1	N	
				:8 Label sensed
				:4,2,1
				0 = Omitted label
				1 = Burroughs standard label
				2 = ANSI standard label
				3 = B 6700 ANSI label
				4 = ANSI label, current MCP
				5 = B 3500 modified ANSI label, MCP and CP
				6 = B 3500 modified ANSI label, MCPV
				7 = LABEL1 installation label
IO-MOD	73	1	N	
				:8 BINARY card input file
				:4 READ with translation
				:2 Status change
				:1 ENABLE allowed for STATUS and I/O error
	74	1	N	Reserved
IO-AUT	75	1	N	Miscellaneous
				:8 Train printer auto train load flag
				:4 SHARED tape flag (Save on CLOSE and H/L)
				:2 Reserved
				:1 Reserved

INPUT/OUTPUT ASSIGNMENT TABLE IO-===
(Continued)

DATA COMMUNICATIONS

	68- 75	8	N	See Standard Devices
IO-DCN	76- 77	2	N	Buffer Number
IO-DCR	78- 79	2	N	Number of I/O Requests on Disk
IO-DCK	80- 81	2	N	Next Disk Request to Read
IO-ACT	82- 87	6	N	Action Label Storage
IO-RDA	88- 93	6	N	Result Descriptor/Address Label Storage
IO-TRN	94	1	N	Translate Table Index
IO-SPO	95	1	N	Remote Capability Level
IO-DC1	96	1	N	
				:8 Dialed Line
				:8/ Leased Line
				:4 Print system message on Remote SPO
				:2 ENABLE/FILL completed
				:1 ENABLE/FILL initiated
IO-DC2	97	1	N	
				:8 Needs overlap message
				:4 Remote SPO waiting log off enable
				:2 Device is on muli-line control
				:1 RJE adaptor type-ASYNC
				:1/ RJE adaptor type-SYNC
IO-DC3	98	1	N	
				:8 Remote SPO
				:4 Data Stream Operation in process
				:2 Halt/Load OCS (SPO) unit
				:1 Reserved
	99	1	N	Reserved

INPUT/OUTPUT ASSIGNMENT TABLE

IO-===
(Continued)

<u>OCS DISPLAY</u>

	60- 81			Reserved
IO-DQ	82- 87	6	N	Last auto message display time (in seconds)
IO-OCS	88- 91	4	N	Address of OCS buffer in KD
	92- 93	2	N	Reserved
	94- 99	6	N	Identical to DCOM IOAT

INPUT/OUTPUT ASSIGNMENT TABLE IO-=== (Continued)

	TAPE			
IO-RL#	76- 78	3	N	Reel Number
IO-CAN	79- 83	5	N	Physical Tape Number
IO-VAR	84	1	N	Density/Parity Variant for Descriptor
IO-MTS	85	1	N	
				:8 Purge after rewind
				:4 Hard ASCII translate (Mod 4 Tape Drive)
				:2 Previous OPEN type: OUT
				:2/ Previous OPEN type: IN
				:1 Scratch tape
IO-MTT	86	1	N	
				:8 Unload tape after rewind
				:4,2,1 Translate type (labels): 0 = No translate required 1 = Internal BCL 2 = External BCL 3 = EBCDIC 4 = 7-bit ASCII 5 = 8-bit ASCII 6 = Reserved 7 = Reserved
IO-PSR	88	1	N	Branch Table Address for Position Return
IO-PSF	89	1	N	
				:8 Reserved
				:4 Ignore tapemarks during position
				:2 Hold overlay during position
				:1 Reverse position
IO-SKP	90- 93	4	N	Skip Block Count on OPEN or Position
IO-PAP	94- 99	6	N	Pseudo-Device Attribute Pointer (Divided by 10)

INPUT/OUTPUT ASSIGNMENT TABLE IO-=== (Continued)

DISK PACK

	68- 75	8	N	Identical to Standard Devices
IO-PSN	76- 81	6	N	Pack Serial Number
IO-FDS	82- 83	2	N	First Directory Sector
IO-FAS	84- 85	2	N	First Available Table Sector
IO-TAS	86- 91	6	N	Total Available Sectors
IO-PEU	92- 93	2	N	SHARED Pack FPM Pseudo EU Number
IO-SHR	94	1	N	SHARED Pack
				:8 Pack is SHARED
				:4 Reserved
				:2 Reserved
				:1 Reserved
IO-FOC	95- 97	3	N	File OPEN Count
IO-PAK	98	1	N	
				:8 Pack not B2000/B3000/B4000 systems generated
				:4 Drive to be powered off
				:2 Restricted pack
				:1 Master pack
IO-DPK	99	1	N	
				:8 Available Table maintenance required
				:4 Reserved
				:2 Reserved
				:1 Reserved

READER/SORTER

	68- 85	18	N	Reserved
IO-SX1	86- 89	4	N	Invalid I/O Count (PCKT-SLCT)
IO-SX2	90- 93	4	N	Invalid I/O Count (ENABLEs)
IO-QAD	94- 99	6	N	Queue Element Address (absolute)

TRAIN PRINTERS

	68- 75	8	N	Identical to Standard Devices
IO-DEF	76- 87	6	A	Default Translator File-ID
	68- 75	8	N	Identical to Standard Devices
IO-LDT	88- 99	6	A	File-ID Presently in Translator

INPUT/OUTPUT ASSIGNMENT TABLE IO-===
(Continued)

DCP

	68- 73	6	N	Reserved
IO-LHR	74	1	N	HALT/LOAD Requirements
				:8 Reserved
				:4 Reserved
				:2 LH required for this DCP
				:1 Warm Load LH permitted
IO-ND#	75	1	N	Number of this DCP
IO-BUF	76- 83	8	N	DCP Entry Address (absolute)
IO-DCF	84	1	N	
				:8 Host output suspended (S-memory full)
				:4 Cancel of default read initiated
				:2 Default read in progress
				:1 Reserved
	85	1	N	Reserved
IO-ERF	86- 87	2	N	Diagnostic Error Code
IO-MEM	88- 91	4	N	Highest S-memory address on DCP in Hexadecimal Words
IO-OPN	92- 93	2	N	Number of MCS files in use on DCP
	94- 95	2	N	Reserved
IO-ERI	96- 99	4	N	Diagnostic Error Information

JOB TABLE JOB-==

Deleted as of ASR 6.7

JOB REFERENCE TABLE JRT-==

Deleted as of ASR 6.7

LABELS LAB===

The file labeling mechanism of the MCP creates the sophisticated hardware/software/systems operator/user program interface necessary to provide all Burroughs customers with unexcelled programming and systems operational ease.

An MCP primary concern is to create an efficient multiprogramming operation where external intervention is held to an absolute minimum and maximum throughput is attained by incorporating automatic file recognition and orderly file creation methods without resorting to a job control language with an inherent confusion factor.

The file labeling techniques used by the MCP provide the system operator with a magnetic tape premount capability (on any tape unit available) so that jobs residing in the schedule will not be delayed upon entering the mix, because of the need to mount a required input file or scratch tape on a specific unit before a given job can commence.

Label handling is a function of the MCP. However, provisions have been made to allow user access to file labels upon input/output through USE routines as specified in the various programming languages.

External Label Formats

Most file types can accommodate external labels. Exceptions on the system include disk and data communications files. The MCP recognizes several label types; unlabeled files with unrecognizable labels are acceptable to the system, though operator intervention is required to achieve file assignment to a program.

Burroughs Standard Label

This label is the standard label for Burroughs B 4000/B 3000/B 2000 Series systems and serves as both the beginning and ending label for all reels of a file (where applicable). Beginning file and beginning reel labels are distinguished by the value of the reel number field. Ending file and ending reel labels are distinguished by a flag in the label field. The label format is given in table 5-3.

Burroughs Standard Label (Continued)

Table 5-3. Burroughs Standard Label Format

This is the standard Burroughs Label Record definition. Note that most fields in the Burroughs Label are alpha mode.

LABELR	0-159			BURROUGHS STANDARD LABEL RECORD
LABELN	0- 15	8	A	" LABEL "
LABZER	16- 17	1	A	Zero
LABMFD	18- 29	6	A	Multifile-ID field; zeros is no MFID
	30- 31	1	A	Blank
	32- 33	1	A	Zero
LABFID	34- 45	6	A	File identifier; blanks for scratch files
	46- 47	1	A	Blank
LABREL	48- 53	3	A	Reel Number for tape files
LABCRE	54- 63	5	A	Creation date (Julian format-YYDDD)
LABCYC	64- 67	2	A	Reserved for cycle; accessible to user to distinguish multiple runs of a program on a single day. Contains 01 by default
LABPGD	68- 77	5	A	Purge date (Julian-YYDDD) date on which MCP may use tape as scratch (if write enabled.)
LABSEN	78- 79	1	A	Sentinel (0 = end-of-file; 1 = end-of-reel; ending label only)
LABBCT	80- 89	5	A	Block count;ending label only
LABRCT	90-103	7	A	Record count; ending label only
LABMDK	104-105	1	A	Beginning Label Memory Dump Key 0 = No memory dump follows label 1 = Memory dump follows label
+LABTPN	106-115	5	A	Physical tape number (insert by operator command)

* TEMPORARY REDEFINE * TEMPORARY REDEFINE *
THE FOLLOWING REDEFINE SHOULD BE ELIMINATED
WHEN THE IOAT IS EXPANDED. JEB 4/18/74

+LABPSN	106-111	6	N	PACK SERIAL NUMBER (OVERFLOW PACK)
LABMSC	116-155	20	A	USER LABEL AREA (PROG ID FOR PRN FILES)
	156	1	N	RESERVED
LABFRM	157	1	N	SPECIAL FORMS FLAG (DISK/TAPE BACKUP) :8 RESERVED :4 RESERVED :2 RESERVED :1 SPECIAL FORMS REQUIRED
	158	1	N	Reserved
LABOOM	159	1	N	OMITTED LABEL FLAG (DISK/TAPE BACKUP) :8 RESERVED :4 IGNORE CHANNEL 12 ON PRINTER FILE (RPG) :2 RESERVED :1 OMITTED LABEL SPECIFIED

Burroughs Standard Label (Continued)

When a label is created, the size may be declared to be greater than the 80 characters described above. Any other space allocated may be employed by the user in any manner desired. The MCP does not interrogate or use any area beyond the normal 80-byte label area.

Certain differences exist between labels created on the various Burroughs systems including recording mode, code, and identifier sizes. The MCP recognizes standard labels created on any of the systems, though identifiers longer than six characters are truncated.

A tape mark is written after the beginning label and before the ending label, and after the last ending label of a physical reel.

USASI Standard Label

The USASI standard label is recognized by the MCP. Further, this label type can be generated by the MCP if such is declared by a program.

The USASI label consists of two or more physical records. The first is a volume record which defines the physical tape reel. (This record is not present in user programs but is recognized or created by the MCP directly.) The format of this record on the system is shown in table 5-4.

Table 5-4. USASI Label Record 1

0- 7	4	A	"VOL1"
8- 9	1	A	Blank
10- 19	5	A	Volume identifier
20-157	69	A	Unused
158-159	1	A	"1"

The second record, following the VOL label is the first file header label. Refer to table 5-5.

Table 5-5. USASI File Header Label Record

0- 7	4	A	"HDR1" for beginning label "EOR1" for end-of-reel label "EOF1" for end-of-file label
8- 19	6	A	File ID
20- 41	11	A	Blank
42- 53	6	A	Multifile ID
54- 55	1	A	0
56- 61	3	A	Reel number

Burroughs Standard Label (Continued)

62- 69	4	A	0001
70- 83	7	A	Blank
84- 93	5	A	Creation date (YYDDD)
94- 95	1	A	Blank
96-105	5	A	Purge date (YYDDD)
106-107	1	A	Blank
108-119	6	A	Ending label block count
120-131	6	A	"MCPV-1"
132-159	14	A	Blank

On tapes from other systems, additional HDR labels may follow the first for example, the B 6700 system creates an HDR2 label which defines the file physical characteristics.

Following the HDR labels are from zero to nine 80-character user header labels (UHL). These labels are user created and contain any data the user wishes to insert.

A tape mark follows the beginning label group, followed by file data which is terminated by a single tape mark.

End-of-file (EOF) labels are placed at the end of a logical file. These records are of the same format as the corresponding HDR records except that EOF replaces HDR. Further, block and record counts are present in the EOF1 record at positions 54 and 60, respectively.

An end-of-reel (EOR) label is placed at the end of a physical reel and is used during reel switching of multi-reel files. This record is in the same format as the EOF record except that EOR replaces EOF.

User trailer records (UTL) may follow either the EOF record(s) or EOR record(s) in a manner analogous to UHL records.

A tape mark follows the ending label group. An additional tape mark follows the last ending label group on a physical reel.

Installation Labels

The MCP provides the capability of recognizing and creating labels defined by the user. This is particularly valuable when tapes from non-Burroughs systems are to be used or created on the system. The label definition is specified in an installation label card which is acceptable input to the CSTRT or WSTRT loaders. All length fields must contain values less than 80. A variable number of labels must not

Burroughs Standard Label (Continued)

be specified unless tape marks are present after both beginning and ending labels. Creation dates must be present in Julian format. One tape mark is expected before the ending label.

All label information must be in the first physical label record. If multiple records are defined, the user is responsible for proper format on output files, and for extraction and verification of any data on input files.

Unlabeled Files

A tape which does not contain a standard, USASI, or installation label is considered to be unlabeled. While such a tape might contain labels recognizable to another system, the MCP considers any tape with unrecognizable initial records to be unlabeled.

Unreadable Labels

If the first records of a tape file cannot be physically read by a tape drive, the tape is said to have an unreadable label. (This is not the same as unrecognizable.) This condition can be due to parity errors in the label records, density or track incompatabilities, or a long blank space on the tape.

Scratch Tape Files

A physical tape reel is considered scratch (usable as an output file) if:

A. The label identifier contains blanks.

B. The current date is equal to or beyond the purge date on the tape and the tape contains a write ring.

C. The tape is unlabeled and contains a write ring.

Case 1, above occurs when the system operator explicitly directs the MCP to purge a tape. Tapes which meet the other criteria are not explicitly purged when detected, but are merely noted as being usable for output if a program requests an output tape.

When a program requests an output tape file, the MCP assigns a tape in the following way. If the relevant FIB is already attached to a tape, that file is OPENed and the

Burroughs Standard Label (Continued)

IOAT is searched for a scratch tape of the same label type declared in the source program. If one can be found it is assigned; otherwise, the first available scratch file is selected.

Program Label Definitions

Four different label specifications are permitted on the system: standard Burroughs system labels, USASI standard labels, installation defined labels, and omitted labels.

The meaning of these specifications depends on the label type and the I/O mode (input or output). For example, a file of any label type is acceptable as input (regardless of the user program specification), but output files always follow the program label specifications.

Within a program, all defined files have at least a minimal label area to hold the file identifier. This area can be extended for certain file types or label declarations.

The relative location of label areas within a program differs with different compilers. For example, the COBOL compiler locates the label immediately after the FIB for the file, while the Assembler locates the label beyond the file buffers. However, the label can always be located from the FIB pointers FIBLAB and FIBLAE. The former addresses the byte prior to the multifile-ID field of the basic label area, and the latter references the byte immediately following the label.

Unlabeled Files

Any file type can be declared unlabeled. The area allocated by the compilers for unlabeled files is a 19-byte extract of the standard Burroughs label and is used to hold the file identifier and reel number (where relevant). While unlabeled tapes do not have a reel number written on the tape, the MCP keeps track of the number of tapes accessed for various purposes. Further, the field is needed if a labeled tape is manually assigned to the requesting program.

An additional byte is allocated beyond this area to make this entire field modulo four in size. This 20-byte label area can be called the Basic Label Area (BLA) and is of the format shown in table 5-7.

Burroughs Standard Label (Continued)

Table 5-7. Basic Label Area

0- 1	1 A	Zero; position to which FIBLAB points
2-13	6 A	Multifile identifier (zeros if no MFID declared
14-15	1 A	Blank
16-17	1 A	Zero
18-29	6 A	File identifier
30-31	1 A	Blank
32-37	3 A	Reel number
38-39	1 A	Filler

The Basic Label Area is the nucleus for all other label types and is accessed by the MCP for file name identification during file OPEN and CLOSE operations and other miscellaneous purposes.

The description is slightly modified for disk files, in that the last 4 bytes are not allocated (total size: 16 bytes). Further, the multifile-ID field is not used for disk files since only a single identifier is permitted.

The label area for a disk pack file is in the standard label format, as described in the following text. However, the fields following the Basic Label Area are not used as they are not relevant for disk pack files.

Standard Labels

The default label specification in source programs is the Standard Label. The memory area allocated is basically constructed by extending the Basic Label Area in both directions to give an area of the size and format of the Standard Label. Table 5-8 shows the format of a Standard Label.

Table 5-8. Standard Label Format

0-15	8 A	" LABEL "
16-55	20 A	Basic Label Area
56-n	52+A	Remainder of Standard Label

The address of the label is calculated by subtracting 16 digits from FIBLAB, the address of the Basic Label Area. The delimiting address is in FIBLAE giving a total area of 80 or more bytes (depending on any user specifications of a user label area). The label addresses in the FIB are used to construct an I/O descriptor for label processing.

Burroughs Standard Label (Continued)

USASI Labels

Regarding the other label types, the USASI Label Area is built around the Basic Label Area. The entire field is shown in table 5-9.

Table 5-9. USASI Label in Program Area

0-15	8 A	" LABEL "
16-55	20 A	Basic Label Area
56-n	80+A	USASI Label formatting area

The first field is not allocated by all compilers (such as, ASMBLR).

The third field is the area into which the USASI label is read on input, or in which output labels are formatted.

If the field is 80 characters in size, only the HDR label can fit the space. More space can be allocated to permit User Header Labels (UHL) or User Trailer Labels (UTL). On input files, after bypassing the VOL label, the MCP reads successive 80-character label records into ascending locations of the program label area until the space is exhausted, a tape mark is sensed, or an unrecognizable record is read (does not contain HDR, UHL, or UTL in the first positions). In the latter case, successive records are read in the same location until a record of the expected type is found or a tape mark is sensed. (Label records can never be read into locations beyond the label area.)

On output, after creating the VOL label, the MCP formats an HDR label in the first 80 characters of the label area and then writes successive 80-character records until the space is exhausted, when a tape mark is written.

CAUTION Sufficient space must be allocated for at least the HDR record or contiguous memory areas may be destroyed when the MCP formats this record.

It is the responsibility of the user program to create UHL and/or UTL records including the correct record identification characters (UHL or UTL) in the first positions.

An example of UHLs in a COBOL68 program follows:

```
          FD   FILENAME
```

Burroughs Standard Label (Continued)

```
          LABEL RECRDS USASI.
      01  LABEL.
      03  HDR-LABEL PICTURE X(80).
      03  UHL-LABEL OCCURS n TIMES.
          05  UHL PICTURE X(3).
          05  UHL-CNT PICTURE 9.
          05  USER-DATA PICTURE X(76).
      DECLARATIVES.
      UHL-LABELER SECTION.  USE AFTER STANDARD BEGINNING LABEL
          PROCEDURE ON FILENAME.
      UHLER.  ADD 1 TO UHL-INDEX.
          MOVE "UHL" TO UHL (UHL-INDEX).
          MOVE UHL-INDEX TO UHL-CNT (UHL INDEX).
          MOVE . . . TO USER-DATA (UHL-INDEX).
          IF UHL-INDEX IS LESS THAN  n GO TO UHLER.
```

NOTE Each record must be exactly 80 characters long; n is the number of UHL labels to be written.

Installation Labels

The label area allocated for an Installation (non-standard) Label is quite similar to that allocated for USASI labels. The Installation Label Area follows the Basic Label Area.

The actual size of installation defined labels is specified to the MCP (through the Installation Label Card); consequently, the compilers are unaware of the actual size and allocate 80 bytes unless a different size is declared. For files with multiple label records, sufficient space must be allocated to hold all records.

For input files, successive label records are read into the Installation Label formatting area until the space or the declared label count is exhausted or a tape mark is sensed. (FIBLAE is used as the delimiting address for all reads; in no case can any contiguous memory area be destroyed.) If memory space is exhausted first, additional records are read into an MCP buffer until sufficient records are bypassed or a tape mark is sensed.

When all label records have been read and/or bypassed, the MCP gives control to the program label USE routine, if present.

For output files, the MCP formats the label record in the first portion of the Installation Label formatting area.

Burroughs Standard Label (Continued)

CAUTION Sufficient space must be allocated for at least one record, or contiguous memory locations may be destroyed when the MCP formats the label record.

If a label USE routine is provided, it is entered at this point. The user then creates any records beyond the initial label record and/or modifies the initial label record. As previously stated, all data specified in the Installation Label Parameter Card is formatted in the first label record by the MCP. If some of the data is to be in other label records, the appropriate information must be moved in the USE routine. However, the data must not be cleared in the first record if the MCP is to recognize the tape label.

After the USE routine is exited, or if none exists, the MCP physically writes the first record. (The size is controlled by the value declared in the Installation Label Parameter Card.)

If multiple labels are defined and space exists in the label area, successive records are written until the label area or the defined count of label records is exhausted. (If a variable number of labels is declared, the MCP sets the label counter to 99, relying solely on the end label address, FIBLAE, and the declared label size to control the number of records to be written.)

Special Cases

Certain tape files created on non-Burroughs systems are not handled automatically by the MCP and special programmatic attention is required. This includes tapes with multiple tape marks before the beginning label, and multifile tapes which are not recognized by the MCP (unrecognizable labels or no labels). This sub-section describes a general method for handling such files; an unlabeled multifile tape will be used as an illustration.

When an unlabeled file is assigned to a program (ULed) and processed by the job, the MCP does not know when EOF occurs. Whenever a tape mark is sensed, the MCP assumes an EOR condition, rewinds the current reel, and requests the next. The operator is responsible for notifying the MCP when no further reels are present (FR message).

The assumption of EOR at each tape mark signifies that straightforward handling of multifile unlabeled tapes is difficult. Such tapes, which have only tape marks between

Burroughs Standard Label (Continued)

the files, can be successfully read to the first tape mark, but at this point, the MCP rewinds the tape. The key to handling such a tape lies in recognizing that the rewind occurs not when the tape mark is physically sensed, but rather when a logical READ is done on the buffer which yielded a tape mark result descriptor.

The programmatic method is to examine the result descriptor (FIBBSW) for the record about to be read. If the descriptor shows a tape mark (C40), the file is CLOSEd NO REWIND to access the next records. The file must have two or more buffers assigned. Thus, a READ is never done on the tape mark buffer and the reel is not rewound.

LOADER PARAMETERS LD-===

The following parameters are passed to the MCP from the Coldsart/Warmstart LOADER deck starting at 959000.

GENERAL LOAD INFORMATION

LD-FLG	0	1 UN	LOAD TYPE INDICATOR
			:8 WARMSTART
			:4 LOAD DEVICE IS TAPE OR PACK
			:2 LOAD DEVICE IS TAPE OR DISK
			:1 COLDSTART
LD-PRC	1	1 UN	PROCESSOR NUMBER FROM MCPDSK CARD (0-3)
LD-MCP	2- 15	6 UA	ID OF MCP TO LOAD

CARD READER INFORMATION

LD-CCH	16-17	2 UN	CARD READER CHANNEL
LD-CFG	18	1 UN	CARD READER FLAGS
			:8 CARD READER ON DLP
			:4 RESERVED (BINARY FLAG)
			:2 RESERVED
			:1 RESERVED (MULTILINE FLAG)
LD-CCI	19-24	6 UN	CARD IMAGE POINTER FOR CARDLESS SYSTEMS

LOAD DEVICE INFORMATION

LD-TCH	25-26	2 UN	LOAD DEVICE CHANNEL (TAPE/DISK/PACK)
LD-TUN	27	1 UN	LOAD DEVICE UNIT (TAPE/PACK)
LD-TFG	28	1 UN	LOAD DEVICE FLAGS
			:8 LOAD DEVICE IS ON DLP (TAPE/DISK/PACK)
			:4 LOAD DEVICE IS BINARY (DISK/PACK)
			:2 RESERVED
			:1 RESERVED (MULTILINE FLAG)
LD-TTR	29	1 UN	TAPE TYPE (TAPE)
			=C GCR
			=4 MPE
			=2 MT9
			=1 MT7
LD-TSZ	30-35	6 UN	COPY TAPE BLOCK SIZE IN DIGITS (TAPE)
LD-THD	36-51	16 UN	HEADER ADDRESS OF FILE BEING LOADED (PACK/DISK)

OCS INFORMATION

LD-OCH	52-53	2 UN	OCS CHANNEL
LD-MUL	54-55	2 UN	OCS MULTILINE CHANNEL OR UNIT
LD-OFG	56	1 UN	OCS FLAGS
			:8 OCS IS ON DLP
			:4 RESERVED (BINARY FLAG)
			:2 RESERVED

LOADER PARAMETERS

LD-=== (Continued)

			:1 OCS IS ON MULTILINE
LD-OTY	57	1 UN	OCS TYPE
			:8 RESERVED
			:4 ODT
			:2 OLD STYLE SPO CONTROL (MS-1)
			:1 TC4000 TYPE SPO
			=0 TD800 TYPE OCS

MCPDSK INFORMATION

LD-DCH	58-59	2 UN	MCPDSK CHANNEL
LD-DP#	60-61	2 UN	MCPDSK PHYSICAL EU (=FF IF NONE SUPPLIED)
LD-DL#	62-64	3 UN	MCPDSK LOGICAL EU
LD-DFG	65	1 UN	MCPDSK FLAGS
			:8 MCPDSK ON DLP
			:4 MCPDSK IS BINARY
			:2 RESERVED
			:1 RESERVED (MULTILINE FLAG)
LD-DTY	66	1 UN	MCPDSK TYPE
			:8 MCPDSK IS LAK DEVICE
			:4 MCPDSK IS 5N DLP
			:2 RESERVED
			:1 RESERVED

FPM INFORMATION

LD-FCH	67-68	2 UN	FPM CHANNEL
LD-FFG	69	1 UN	FPM TYPE
			:8 FPM IS ON DLP
			:4 RESERVED (BINARY FLAG)
			:2 RESERVED
			:1 RESERVED (MULTILINE FLAG)
	70-99	32 UN	RESERVED

LABEL FILE RECORD FORMATS LT====

Label Type Index Header

```
                                 (8 BIT)
LTH       0-39               LABEL INDEX HEADER ENTRY
          0- 7      8 N      RESERVED
LTH-#T    8- 9      2 N      NUMBER OF LABEL TYPES DECLARED
LTHMRL   10-13      4 N      MAXIMUM REQD LENGTH OF ANY TAPE LABEL
LTHSTB   14-17      4 N      SEGMENT OFFSET TO STATUS TEMPLATES
LTHSTL   18-21      4 N      LENGTH OF STATUS TEMPLATES IN DIGITS
         22-39     18 N      RESERVED
```

Label Type Index Table Entry

There is one of these records for each label type declared on the system.

```
LIT       0-39               LABEL TYPE ENTRY
LT-NAM    0-15      8 A      SYMBOLIC LABEL NAME (8 BYTES)
LT-OCD   16-19      4 N      OPEN-CLOSE DESCRIPTION BLOCK SEGMENT ADD
LT-OCL   20-21      2 N      OPEN-CLOSE DESCRIPTION BLOCK LENGTH-SEGMENTS
         22-29      8 N      RESERVED
LT-USR   30-31      2 N      OPEN-CLOSE BLOCK IN-USE COUNT
LT-ADR   32-39      8 N      OPEN-CLOSE BLOCK MEMORY ADDRESS
```

MCP OVERLAY REQUEST TABLE M-====

M-OVRQ	0-49			MCP OVERLAY REQUEST QUEUE ENTRY
M-LINK	0- 3	4	N	LINK TO NEXT MCP OVERLAY REQUEST ENTRY
M-MIX	4- 5	2	N	MIX NUMBER OF PROGRAM REQUESTING MCP OVERLAY
M-RTRN	6-11	6	N	GLOBAL RETURN ADDRESS
M-LGH	12-13	2	N	LENGTH OF PARAMETERS FOLLOWING "NTR"
M-PARM	14-49	36	N	PARAMETERS PASSED TO MCP OVERLAY VIA "NTR"

```
          MEMORY ALLOCATION TABLE                                MAP-==

     MAP        0-11             MEMORY ALLOCATION TABLE ENTRY
     MAP-AD     0- 3    3 N      ADDRESS OF MEMORY BLOCK IN KD

     MAP-TP        4    1 N      MAP ENTRY TYPE
                                 0 = AVAIL MEMORY ENTRY
                                 1 = MISCELLANEOUS PUSHABLE MEMORY BLOCK
                                     MAP-SU = 0,  STOQUE BLOCK
                                     MAP-SU = 1,  DCOM TRANSLATE TABLE
                                     MAP-SU = 2,  MCS QUEUE
                                     MAP-SU = 3,  ADDRESS BLOCK
                                     MAP-SU = 4,  SECURITY BUFFER
                                 2 = PUSHABLE PROGRAM ENTRY
                                 3 = IOAT, HEADER, HEADER ADDRESS BLOCK (KD)
                                 4 = PSEUDO RDR, PBD, OR PBT BLOCK (2 KD)
                                     MAP-SU = 0,  PSEUDO DEVICE ENTRY
                                     MAP-SU = 1,  MCS EXTERNAL BUFFER
                                     MAP-SU = 7,  MCP/LIO PORT AND SUBPORT
                                     MAP-SU = 8,  SUBPORT BLOCK
                                     MAP-SU = 9,  PORT BLOCK
                                 5 = MISCELLANEOUS NON-PUSHABLE MEMORY BLOCK
                                     MAP-SU = 0,  GENERAL PURPOSE GARBAGE
                                     MAP-SU = 1,  REMSPO BUFFER BLOCK
                                     MAP-SU = 2,  OCS OUTPUT BUFFER
                                     MAP-SU = 3,  OCS INPUT BUFFER
                                     MAP-SU = 4,  MLOG BUFFER BLOCK
                                     MAP-SU = 5,  PROCESSOR SNAP MAP BLOCK
                                 6 = NON-PUSHABLE PROGRAM ENTRY
                                     (MICR/DCOM FILES)
                                 7 = EXTENSION  MODUAL ALLOCATION
                                 8 = QUIKMEM CARD ENTRY
                                 9 = XM MEMORY ENTRY
                                 A = TIME-SHARING MEMORY BLOCK
                                 C = MCP HI MEMORY TABLES
                                 D = GLOBAL ENTRY
                                 E = TRAILER ENTRY
                                 F = NULL ENTRY

                    FORMAT FOR ALL TYPES EXCEPT TYPE 3

     MAP-SZ     5- 8    4 N      SIZE OF MEMORY BLOCK IN KD
     MAP-MX     9-10    2 N      MIX # OF USER
     MAP-SU       11    1 N      SUPPLEMENTARY TYPE

                    TYPE 3 FORMAT

     MAP-FL     5- 9    5 N      :8/4 0/0 = AVAILABLE FOR ASSIGNMENT
                                      1/0 = IOAT/DISK HDR BLK - 200 DIGITS
                                      1/0 = AVAILABLE FOR ASSIGNMENT
                                      1/1 = AVAILABLE FOR ASSIGNMENT
```

MEMORY ALLOCATION TABLE MAP-==
(Continued)

```
:2/1 1/0 = 2ND 100 DIGITS IN USE (2 BIT ONLY)
     0/1 = 1ST 100 DIGITS IN USE (1 BIT ONLY)
     1/1 = 1ST & 2ND 100 DIGITS IN USE
           (BOTH BITS)
           EACH DIGIT CORRESPONDS TO SUCCESSIVE
           200 DIGITS IN 1 KD BLOCK
```

MAILBOX MB-===

MB-BOX	0-199			MAILBOX RECORD
MB-INP	0	1	N	0:8 MAILBCX IN-USE FLAG
				0:4 RESPONSE DEMAND FLAG
				0:2&1 PROCESSOR NUMBER OF INITIATING SYSTEM
	1- 3	3	N	AVAILABLE
MB-CM0	4- 5	2	N	COMMAND TO PROCESSOR #0
MB-RS0	6- 7	2	N	RESPONSE FROM PROCESSOR #0
MB-CX0	8- 19	2	N	EXTENSION COMMAND TO PROCESSOR #0
MB-RX0	20- 31	2	N	EXTENSION RESPONSE FROM PROCESSOR #0
MB-CM1	32- 33	2	N	COMMAND TO PROCESSOR #1
MB-RS1	34- 35	2	N	RESPONSE FROM PROCESSOR #1
MB-CX1	36- 47	12	N	EXTENSION COMMAND TO PROCESSOR #1
MB-RX1	48- 59	12	N	EXTENSION RESPONSE FROM PROCESSOR #1
MB-CM2	60- 61	2	N	COMMAND TO PROCESSOR #2
MB-RS2	62- 63	2	N	RESPONSE FROM PROCESSOR #2
MB-CX2	64- 75	12	N	EXTENSION COMMAND TO PROCESSOR #2
MB-RX2	76- 87	12	N	EXTENSION RESPONSE FROM PROCESSOR #2
MB-CM3	88- 89	2	N	COMMAND TO PROCESSOR #3
MB-RS3	90- 91	2	N	RESPONSE FROM PROCESSOR #3
MB-CX3	92-103	12	N	EXTENSION COMMAND TO PROCESSOR #3
MB-RX3	104-115	12	N	EXTENSION RESPONSE FROM PROCESSOR #3
MB-EUS	116-155	40	N	EUSTAT-S OF SHARED DISK EU-S
MB-S#1	156-157	2	N	NUMBER OF SHARED EU-S ON DISK SUBSYSTEM #1
MB-S#2	158-159	2	N	NUMBER OF SHARED EU-S ON DISK SUBSYSTEM #2
MB-S#3	160-161	2	N	NUMBER OF SHARED EU-S ON DISK SUBSYSTEM #3
MB-FLG	162-164	3	N	COPY OF SUBFLG DISK SUBSYSTEM STATUS
MB-FPM	165-168	4	N	NUMBER OF FILE PROTECT WORDS ON PRIMARY SUBSY
	169-199	31	N	AVAILABLE

MIX TABLE MIX-==

The MIX table (MIX) is used by the MCP to maintain information about all programs.

Among the information stored in the MIX table are program location in memory, program address, wait indicators (why a program cannot continue), various times, and program status. The eight digits of the MIX entry, starting with MIX-OV, determine the reinstateability of a program. If no bits are on in these eight digits, the program can be reinstated. If any bit is on, the program is awaiting MCP or operator action.

A copy of the MIX table is stored on disk and is used to print out information about what jobs existed prior to a halt/load if the memory version of the MIX table has been corrupted. The MCP is assigned a MIX number of zero.

The following describes the fields in the MIX table. The information is used by the MCP in maintaining a job while executing in the system.

The MIX Table consists of a link field plus ten parts.

The first part consists of wait flags. If any of the wait flags are non-zero the program will not be able to run.

The second part is the program identification information. These fields may be searched to find particular programs.

The third part is the information used during scheduling. This includes a status digit, a precedence link and value data.

The fourth part is the program reinstate information. It consists of the base, limit, PAR, accumulator, etc. This information is used both by the hardware to actually reinstate the job, and by the MCP for determining the address and validity of program-relative parameters. Note: This section must be at a mod-4 address.

The fifth part is supplementary wait information used to further identify a waiting condition (hardware type waiting for, etc.)

The sixth part is program status information. This includes file counts, terminate and DS codes, and numerous flags.

MIX TABLE MIX-==
(Continued)

The seventh part is timing information. This consists of current processor, waiting I/O, prorate, stopped and average RUN/WAIT time and the time limit.

The eight part is security information such as usercode, charge number, and security level and class.

The ninth part is special file information, including disk addresses for the code file, insert file, label equate file, stopped file and roll-out file. Note: This section must be at a mod 4 address.

The tenth part is information needed if this is a timesharing job.

```
                        NORMAL STATE PROGRAM MIX TABLE FORMAT

MIX-LK   0-   5     6 N   LINK FOR SLL
         6-   7     2 N   <<RESERVED TO EXPAND LINK>>

                          PART 1 - WAIT FLAGS.

MIX-OV        8     1 N   WAIT COUNT FOR MCP OVERLAY AREA

MIX-IO        9     1 N   WAITING I/O PROCESSING
                           1 = READ/WRITE IOC
                               (ALSO DCOM FILE IOC)
                           2 = FILE CLOSE QUEUE FLUSH
                           3 = STOPPAGE
                           4 = NO SPACE IN COMPLEX WAIT TABLE
                           5 = POSITIONING
                           6 = DATA COMM FILL/ENBL (WAIT STATUS)
                           7 = PROGRAM OVERLAY
                           8 = COMPLEX WAIT
                           9 = TERMINATE QUEUE FLUSH
                           A = WTG LIO OPEN
                           B = WTG LIO CLOSE
                           C = WTG LIO ATTRIBUTE INTERROGATE

MIX-WM       10     1 N   WAITING MODULE PROCESSING
                           1 = DATA COMM FOR BCT PROCESSING
                           2 = CORE-TO-CORE SEND/RECEIVE
                           3 = WAITING BNAM MODULE
                           4 = STOQUE DATA ENTRY
                           5 = STOQUE MEMORY
                           6 = STOQUE NAME SLOT
                           7 = STOQUE FOR PROCESSING
                           8 = WAITING TRACE
```

```
          MIX TABLE                                         MIX-==
                                                                (Continued)

                                A = TSM FOR HANDLER TRANSFER
                                B = WAITING DCPC

      MIX-OK        11      1 N  WAITING OPERATOR/MCP ACTION
                                1 = DUPLICATE FILE (NON-DISK)
                                2 = DUPLICATE LIBRARY ON DISK
                                    (OR PACK CLOSE)
                                3 = NO USER DISK OR PACK
                                4 = NO FILE ON DISK OR PACK
                                5 = LOCKED FILE ON DISK OR PACK
                                6 = NO FILE (NON-DISK)
                                7 = OUTPUT DEVICE REQUIRED
                                8 = EXTENSION MODULE NOT IN MEMORY
                                    (MIX-HW INDICATES MODULE WTG FOR)
                                     MIX-HW = 1 - DCOM
                                     MIX-HW = 2 - MICR
                                     MIX-HW = 3 - CRCR
                                     MIX-HW = 4 - STOQ
                                     MIX-HW = 5 - DCP
                                     MIX-HW = 6 - RESERVED
                                     MIX-HW = 7 - RESERVED
                                     MIX-HW = 8 - DMS2
                                9 = WAITING DMS FUNCTION
                                A = LOCAL SPO ACCEPT
                                B = REMOTE SPO ACCEPT
                                C = WAITING OPEN HARDWARE
                                D = WAITING MISSING PACK
                                E = WTG BLOCK COUNT ERROR ACTION
                                F = ZIP/SPOM PROCESSING

      MIX-CR        12      1 N  WAITING MEMORY
                               :8  WTG TRACE BACKUP MEMORY
                                   (MAY BE SET WITH OTHER VALUES)
                                0 = NOT WAITING (OR TRACE)
                                1 = DSK/BKP OPEN
                                2 = DIRECT I/O OPEN
                                3 = DCP OPEN
                                4 = DCOM OPEN
                                5 = SHARE MIX PIB PROC OPEN
                                6 = PORT/SUBPORT OPEN OR SET

      MIX-WA        13      1 N :8 PUSH IN PROCESS
                               :4  PROGRAM STOPPED (SEE MIX-HI & MIX-HO)
                               :2  WAITING IOC ON MICR/OCR
                               :1  SHARED AREA WAITING RUN
                               :1  TIME-SHARING PROCESS WTG I/O QUEUE FLUSH

      MIX-WB        14      1 N :8 PROGRAM SLEEPING (DOZE OR DCOM WAIT)
```

```
          MIX TABLE                                          MIX-==
                                                                (Continued)

                                       :4 TRACE PRINT IN PROCESS
                                       :2 TRACE PRINTER REQD
                                       :1 TRACE DISK REQD

   MIX-WC        15      1 N   :8 TIME SHARING PROCESS WTG TERMINAL IOC
                               :4 PROGRAM SUSPENDED
                               :2 TIME SHARING PROCESS WAITING SWAPPING
                               :1 WAITING DIRECT IOC

   MIX-SP        16      1 N   SCHEDULE PRIORITY
   MIX-MP        17      1 N   MEMORY PRIORITY
   MIX-RP        18      1 N   PROCESS PRIORITY
   MIX-IP        19      1 N   I/O PRIORITY
   MIX-Q#        20      1 N   TIME-SHARING SCHEDULING QUEUE #
   MIX-QP        21      1 N   PRIORITY WITHIN TIME-SHARING SCHEDULING
                               QUEUE
   MIX-PN    22- 23      2 N   POSITION IN PRIORITY QUEUE (TIME SHARING)
             24- 25      2 N   RESERVED

                               PART 2 - PROGRAM IDENTIFICATION.
                                        (SEARCHABLE FIELDS)

   MIX-ID    26- 37      6 A   PROGRAM ID
   MIX-MF    38- 49      6 A   MULTIPROGRAM ID
   MIX-RQ    50- 51      2 N   DAT/MIX/PCR REQUESTOR CODE
             52- 53      2 N   <<RESERVED FOR REQUESTOR EXPANSION>>
   MIX-RJ    54- 55      2 N   RJE ORIGINATOR KEY
             56- 57      2 N   <<RESERVED FOR ORIGINATOR EXPANSION>>
   MIX-RL    58- 61      4 N   RUN LOG ID NUMBER

   MIX-PI        62      1 N   PROGRAM INITIATE CODE
                                   0 = EXECUTE
                                   1 = COMPILE (COMPILE PHASE)
                                   2 = COMPILE SYNTAX
                                   3 = COMPILE LIBRARY
                                   4 = COMPILE (EXECUTE PHASE)
                                   5 = RUN
                                   6 = SHARE
                                   7 = COMPILE SAVE (COMPILE PHASE)
                               :8 EXECUTE PHASE TO BE SCHEDULED
                                  (TERMINATE)

   MIX-PG        63      1 N   SPECIAL PROGRAM CODE
                                   1 = PROGRAM IS A GENERATOR
                                   2 = PROGRAM IS DMPALL
                                   3 = HEAD-PER-TRACK LOADMP
                                   4 = DISK PACK LOADMP
                                   5 = PROGRAM IS DSKOUT OR PACK SQUASH
```

				6 = PROGRAM HAS DCP MCS STATUS
				7 = TIME SHARING PROCESS
				8 = TIME SHARING MAIN MIX ENTRY
				9 = CLEAR SENSITIVE DATA PGM (CLRSNS)
				A = GENERATOR IN SHARED AREA
				B = DMS CONTROL PROGRAM
				C = WFL HANDLER
				D = BNA HANDLER
				E = PROGRAM IS COPY
				PART 3 - SCHEDULE INFO.
MIX-SS	64	1	N	SCHEDULE STATUS DIGIT
				0 = JOB IN MIX
				A = SCHEDULE COMPLETE (READY FOR MIX)
				B = SCHEDULE COMPLETE (PRECEDENCE LINK)
				E = SCHEDULE IN PROCESS
				F = AVAILABLE SCHEDULE SLOT
MIX-PL	65- 66	2	N	PRECEDENCE LINK (JOB NO OF PROGRAM OR ZERO)
	67- 68	2	N	<<RESERVED TO EXPAND LINK>>
MIX-VA	69- 70	2	N	VALUE ADDRS (INTRN ONLY, LOW ORDER DIGITS)
MIX-VL	71	1	N	VALUE LENGTH
MIX-VD	72- 79	8	N	VALUE DATA
MIX-GT	80- 99	20	N	TRACE PARAMETERS (DURING SCHEDULE)
				PART 4 - REINSTATE INFORMATION.
MIX-RN	100	1	N	PROGRAM EXECUTION FLAGS (B4800)
MIX-RN				:8 USER PROGRAM FLAG - RUN LIGHT
MIX-RN				:4 SNAP GATE ENABLE FLAG
MIX-RN				:2 BCT OVERRIDE FLAG - ALWAYS OFF
MIX-RN				:1 (<<RESERVED>>)
MIX-BH	101	1	N	HIGH-ORDER DIGIT OF BASE REGISTER (B4800)
MIX-LH	102	1	N	HIGH-ORDER DIGIT OF LIMIT REGISTER (B4800)
MIX-PA	103-109	7	N	PROGRAM ADDRESS REGISTER
MIX-BL	110-112	3	N	LOW-ORDER DIGITS OF BASE REGISTER
MIX-LL	113-115	3	N	LOW-ORDER DIGITS OF LIMIT REGISTER
MIX-TG	116	1	N	MODE/OVERFLOW/CONDITION TOGGLE
MIX-HE	117	1	N	HALT EXECUTION/ASCII TOGGLE
MIX-NO	118-119	2	N	JOB NUMBER
	120-123	4	N	<<RESERVED FOR JOB NUMBER EXPANSION>>
MIX-AC	124-143	20	N	FIXED ARITHMETICS ACCUMULATOR STORAGE
MIX-BA	144-147	4	N	BASE ADDRESS
MIX-LM	148-151	4	N	LIMIT ADDRESS

```
        MIX TABLE                                            MIX-==
                                                          (Continued)


                                PART 5 - SUPPLEMENTARY WAIT INFORMATION.

    MIX-XX   152-157     6 N    IOAT ADDRESS OF WORKING FILE
                                WAITING MEMORY:
                                  FOR STOQ  - ENTRY SIZE IN DIGITS (6UN)
                                  ALL OTHER - MEMORY REQUIRED IN KD (4UN)
                                EXPAND/CONTRACT PROGRAM MEMORY SIZE AS:
                                  SXXXFO  WHERE: S = SIGN +/- FOR ADD/SUB
                                               XXX = MEM CHANGE SIZE IN KD
                                                 F = HEX "F" FOR VERIFY
                                WAITING DCP:  DCP NUMBER  (1 UN)
    MIX-YY   153-157            TIME TO WAKE UP FROM DOZE
                                TIME WAITING IOC INITIATED

    MIX-IH   158-159     2 N    DATA COMM WAIT POINTER (IO-STA OF NEXT UNIT)

    MIX-HW   160-161     2 N    HARDWARE TYPE WAITING FOR
                                EXTENSION MODULE WAITING FOR (MIX-OK = 8)
                                 01 = DCOM
                                 02 = MICR
                                 03 = CRCR
                                 04 = STOQ
                                 05 = DCP
                                 06 = ????
                                 07 = PACK
                                 08 = DMS2

    MIX-HI       162     1 N    MIX-HO VALUE WHEN HIHO IS WAITING DISK/MEMORY
    MIX-HO       163     1 N    HIHO STATUS DIGIT
                                 1 = STOP IN PROCESS: KBD/ZIP STOP RQST
                                 2 = STOP IN PROCESS: PRIORITY CRASHOUT
                                 3 = STOP IN PROCESS: PUSH INITIATED
                                 4 = STOP IN PROCESS: SORT ROLL-OUT
                                 5 = STOP IN PROCESS: BREAKOUT RQST
                                 6 = STOP IN PROCESS: MEM DUMP TO DISK RQST
                                 7 = SORT SPECIFICATION FILE BUILD IN
                                     PROCESS
                                 8 = PROGRAM IS KBD/ZIP STOPPED
                                 9 = PROGRAM IS PRIORITY STOPPED
                                 A = PUSH ROLL-IN PHASE IN PROCESS
                                 B = SORT ROLL-IN PHASE IN PROCESS
                                 C =
                                 D =
                                 E = WAITING DISK
                                 F = WAITING MEMORY

                                PART 6 - PROGRAM STATUS INFORMATION.
```

MIX TABLE

MIX-==
(Continued)

```
MIX-IC   164-166      3 N NO. OF IOATS ASSIGNED TO PROGRAM
MIX-FC   167-169      3 N NUMBER OF FILES DECLARED
MIX-SG   170-175      6 N DISK SEGMENTS IN PROGRAM
MIX-TC   176-177      2 N TERMINATE CODE
                          00   NORMAL EOJ
                          01   ADDRESS ERROR
                          02   INVALID I/O LIMIT
                          03   INVALID OPEN
                          04   INVALID CLOSE
                          05   INVALID READ
                          06   INVALID WRITE
                          07   EOF NO LABEL
                          08   PARITY NO LABEL
                          09   OVERLAY I/O ERROR
                          10   INVALID INSTRUCTION
                          11   OPERATOR DS
                          12   DCOM I/O ERROR
                          13   INVALID I/O DESCRIPTOR
                          14   MEMORY PARITY ERROR
                          15   INSTRUCTION TIMEOUT
                          16   INVALID DCOM I/O REQUEST
                          17   SYNTAX ERROR (COMPILER)
                          18   ABORTED - HALT LOAD
                          19   INVALID CORE TO CORE REQUEST
                          20   INVALID STOQUE REQUEST
                          21   TIME LIMIT EXCEEDED
                          22   STALEMATE - NO LABEL
                          23   NO SORT MEMORY
                          24   INVALID BCT PARAMETER
                          25   INVALID FILE POSITION
                          26   TAPE LABEL ERROR
                          27   INVALID OPERATOR MAG TAPE REWIND
                          28   LOST HANDLER  RJE/WFL/BNA
                          29   FILE SECURITY ERROR
                          30   INV FILE ATTRIBUTE SET REQUEST
                          31   INV FILE ATTRIBUTE GET REQUEST
                          32   INV PORT FILE I/O REQUEST
                          33   BNA ERROR
                          34   DMS ERROR
MIX-DS   178-179      2 N USER DS CODE (FROM DS/DP)
         180-181      2 N RESERVED

MIX-FP       182      1 N FILE PARAMETER BLOCK FLAG (PB-FPF)
                              0 = NO FILE PARAMETER BLOCKS
                              1 = 100 DIGIT FILE PARAMETER BLOCKS
                              2 = 200 DIGIT FILE PARAMETER BLOCKS
                              3 = 200 DIGIT FILE PARAMETER BLOCKS
```

```
                    MIX TABLE                                      MIX-==
                                                              (Continued)

                                    4 = 200 DIGIT NEW FILE PARAMETER BLOCKS
                                    5 = 200 DIGIT NEW FPBS AND PORT FILES


         MIX-DD      183      1 N :8 INITIATED THROUGH ZIP COMMUNICATE
                                  :4 EXECUTE WITH LOCK
                                  :2 INITIATED FROM PSEUDO CARD READER
                                  :1 CHARGE NUMBER SUPPLIED

         MIX-LQ      184      1 N :8 MEMORY DUMP RQST ON ABNORMAL TERMINATION
                                  :4 INITIATED FROM A REMOTE SPO
                                  :2 TIME LIMIT SUPPLIED
                                  :1 LABEL EQUATE SUPPLIED

         MIX-SR      185      1 N :8 NO PUSHDOWN/STOP
                                  :4 BREAKING OUT
                                  :2&1 SORT/BREAKOUT PROGRAM INDEX
                                     0 = USER PROGRAM
                                     1 = SORT.V  (DISK SORT)
                                     3 = SORT: OR MERG:  (6.4+ SORT OR MERGE)

         MIX-DC      186      1 N :8 SIMULATE PROCESSOR INTERRUPT IN PROG MEM
                                  :4 MIDNIGHT OVERLAP FOR DOZE
                                  :2 DISALLOW OPERATOR BREAKOUT REQUESTS
                                  :1 USE PROCEDURE IN PROCESS

         MIX-TR      187      1 N :8 TERMINATE RUNNING
                                  :4 CALL TERMINATE AT COMPLETION OF
                                     DUMP/BRKOUT
                                  :2 PASS BREAKOUT R/D'S TO PROG IF ARMED
                                  :1 BREAKOUT NOT ALLOWED - PACK OR DIRIO FILE

         MIX-FL      188      1 N :8 PREVIOUS ACPT & DISPLAY MSGS ON DISK.
                                  :4 CODE FILE IS ON DISKPACK
                                  :2 RJE ORIGINATED JOB (HANDLER LEFT)
                                  :1 COMPILING TO PACK FLAG

         MIX-PR      189      1 N :8 PARAMETERS PASSED
                                  :4 RESERVED
                                  :2 SAVE OVLY IN VS (MCPIX ONLY)
                                  :1 RESERVED

         MIX-FG      190      1 N :8 BOUND FLAG (MCP INTRINSIC)
                                  :4 TEST FLAG
                                  :2 DEBUG FLAG
                                  :1 RERUN FLAG

         MIX-TP      191      1 N :8 TRACING PGM (OVERLAYABLE TRACE)
```

MIX TABLE

MIX-==
(Continued)

				:4 TRACE HEADING REQUIRED
				:2 TSM MAIN MIX (MIX-PG = 8)
				:1 CLOSE IN PROCESS
MIX-TM	192	1	N	:8 SET TRACE AT BOJ
				:4 MCP WORKFILE IN USE BY PROGRAM (%MXOPD)
				:2 NO CODE FILE AT TERMINATE
				:1 SYNTAX ERROR
MIX-FF	193	1	N	:8 USER PROGRAM MUST BE USED
				:4 MCP INTRINSIC MUST BE USED
				:2 GO-PHASE (CMP & GO) LBL EQUATE DSK OBTAINED
				:1 CMP/EXECUTE PHASE LBL EQUATE DSK OBTAINED
MIX-EU	194-195	2	N	DISK EU FOR GENERATOR CODE FILE
MIX-CF	196-207	6	A	FAMILY NAME FOR GENERATED CODE FILE

PART 7 - TIMING INFORMATION.

MIX-TL	208-212	5	N	TIME LIMIT (REMAINING SECONDS)
MIX-RT	213-216	4	N	ACCUMULATED RUN TIME THIS N-SECOND
MIX-WT	217-220	4	N	ACCUMULATED WAIT IOC TIME THIS N-SECOND
MIX-DT	221-228	8	N	ACCUMULATED DIRECT TIME
MIX-PT	229-236	8	N	ACCUMULATED PRO-RATED TIME
MIX-IT	237-244	8	N	ACCUMULATED WAITING IOC TIME
MIX-AV	245-248	4	N	AVERAGE RUN/WAIT TIME
MIX-BJ	249-253	5	N	BEGINNING OF JOB TIME IN SECONDS
MIX-TI	254-258	5	N	TOTAL STOPPED TIME IN SECONDS

PART 8 - SECURITY INFORMATION.

MIX-ET	259	1	N	:8 USRTBL ENTRY HAS BEEN MADE
				:4 USERFL MAINTENANCE POSSIBLE
				:2 USER CARD ENTRY MADE FOR THIS PROGRAM
				:1 RESERVED FOR SYAC
MIX-UC	260-279	10	A	INITIATOR'S USERCODE
MIX-LV	280	1	N	INITIATOR'S CAPABILITY (SPO) LEVEL
MIX-PV	281	1	N	:8 USER CAN DO LIBMAINT TO OTHER USER'S
				:4 USER CAN DO DIRECT I/O FILES
				:2 RESERVED
				:1 RESERVED

MIX TABLE

MIX-==
(Continued)

	282	1	N	RESERVED	
MIX-CL	283	1	N	SECURITYTYPE	
				8 CONTROLLED	
				4 GUARDED	
				2 PUBLIC	
				1 PRIVATE	
				F NONE	
MIX-CN	284-289	6	N	CHARGE NUMBER	
MIX-SU	290	1	N	GENERATED PROGRAM'S SECURITYUSE	
MIX-SN	291	1	N	" " SENSITIVEDATA FLAG	
MIX-GD	292-303	6	A	" " SECURITYGUARD	
MIX-FA	304-315	6	A	" " SECURITYFAMILY	

PART 9 - SPECIAL FILE INFORMATION.

MIX-PB	316-331	16	N	PGM PARAMETER BLK DISK ADDRESS	
MIXPBE	317-319	3	N	PGM PARAMETER BLK EU #	
MIXPBA	320-331	12	N	PGM PARAMETER BLK ADDRESS	
MIX-IN	332-347	16	N	DISK ADDRESS OF 1ST INSERT (OR VALUE) SEG	
MIXINE	333-335	3	N	1ST INSERT SEG EU #	
MIXINA	336-347	12	N	1ST INSERT SEG ADDRESS	
MIX-FH	348-363	16	N	DISK FILE HDR ADDRESS	
MIXFHE	349-351	3	N	DISK FILE HDR EU #	
MIXFHA	352-363	12	N	DISK FILE HDR DISK ADDRESS	
MIX-FD	364-379	16	N	DISK DIRECTORY HDR ADDRESS	PROGRAM FILE
MIXFDE	365-367	3	N	DISK DIRECTORY EU #	
MIXFDA	368-379	12	N	DISK DIRECTORY DISK ADDRESS	
MIX-FX	380-383	4	N	INDEX INTO DIRECTORY HDR	
MIX-EH	384-399	16	N	DISK FILE HDR ADDRESS	
MIXEHE	385-387	3	N	DISK FILE HDR EU #	
MIXEHA	388-389	12	N	DISK FILE HDR DISK ADDRESS	
MIX-ED	400-415	16	N	DISK DIRECTORY HDR ADDRESS	LABEL EQUATE FILE
MIXEDE	401-403	3	N	DISK DIRECTORY EU #	
MIXEDA	404-415	12	N	DISK DIRECTORY DISK ADDRESS	
MIX-EX	416-419	4	N	INDEX INTO DIRECTORY HDR	
MIX-SH	420-435	16	N	DISK FILE HDR ADDRESS	
MIXSHE	421-423	3	N	DISK FILE HDR EU #	
MIXSHA	424-435	12	N	DISK FILE HDR DISK ADDRESS	
MIX-SD	436-451	16	N	DISK DIRECTORY HDR ADDRESS	STOPPED FILE
MIXSDE	437-439	3	N	DISK DIRECTORY EU #	
MIXSDA	440-451	12	N	DISK DIRECTORY DISK ADDRESS	

```
          MIX TABLE                                          MIX-==
                                                               (Continued)

     MIX-SX  452-455      4 N INDEX INTO DIRECTORY HDR

     MIX-RH  456-471     16 N DISK FILE HDR ADDRESS
     MIXRHE  457-459      3 N DISK FILE HDR EU #
     MIXRHA  460-471     12 N DISK FILE HDR DISK ADDRESS
     MIX-RD  472-487     16 N DISK DIRECTORY HDR ADDRESS   ROLL-OUT FILE
     MIXRDE  473-475      3 N DISK DIRECTORY EU #
     MIXRDA  476-487     12 N DISK DIRECTORY DISK ADDRESS
     MIX-RX  488-491      4 N INDEX INTO DIRECTORY HDR

     MIX-BR  492-497      6 N PREVIOUS BREAKOUT DISK FILE NUMBER
     MIX-NM  498-509      6 A BREAKOUT TAPE/PACK NAME
     MIX-BK  510-511      2 N BREAKOUT PARAMETERS

                              PART 10 - TIMESHARING INFORMATION.

     MIX-AL  512-516      7 N ACTION LABEL IN TIME-SHARING HANDLER
     MIX-BF  519-525      7 N BUFFER ADDRESS IN TIME-SHARING HANDLER
     MIX-MA  526-529      4 N MEMORY ADDRESS OF USER DATA AREA
     MIX-US  530-533      4 N SIZE OF USER DATA AREA IN KD
     MIX-CS  534-537      4 N SIZE OF CODE FILE
     MIX-MM  538-539      2 N MIX # OF MAIN MIX ASSOC. WITH THIS PROCESS
             540-541      2 N <<RESERVED FOR MIX NUMBER EXPANSION>>
     MIX-P#  542-543      2 N PROCESS NUMBER IF PROCESS IS IN PIB
     MIX-PS  544-545      2 N SIZE OF USER DATA AREA PROCESS STACK

     MIX-CC      546      1 N :8 JOB HAS NOT RUN FOR ONE N-SECOND PERIOD
                              :4 FORCE ROLLIN FROM DISK
                                 (THIS FLAG IS SET BY PROCESS CALL IF
                                 PARAMS HAVE BEEN PASSED AND THE
                                 PROCESS IS IN THE PIB BUT THE PROCESS
                                 IS NOT TYPE II.  THIS  KEEPS OTHERS
                                 FROM GETTING OUR INSERTS.  THIS FLAG
                                 CAUSES THE MIX-D1 DISK AREA TO BE
                                 USED INSTEAD OF PIB-DA).
                              :2 BEGINNING OF PROCESS
                                 (SET BY PROCESS CALL, TYPE CONVERSION,
                                  AND ABNORMAL E-O-P OF THE BASE PROCESS).
                              :1 PROCESS IS NOT IN PIB (USER CREATED)

     MIX-ST      547      1 N :8 JOB OR PROCESS TO BE TERMINATED
                              :4 RESERVED
                              :2 USER HAS BEEN SUSPENDED
                              :1 IF TMX-XX:8 = 1 AND
                                 SET: JOB TO BE DS-ED
                                 RESET: CURRENT PROCESS TO BE TERMINATED

     MIX-CD      548      1 N :8 DMSII JOB NEEDS TO BE MADE AVALIABLE
```

MIX TABLE MIX-==
(Continued)

				:4 LOW MEMORY SHOULDN'T BE INITIALIZED (SET BY TYPE CONVERSION ONLY).
				:2 CODE FILE IS TYPE II PROCESSOR
				:1 CODE FILE IN MEMORY
MIX-UD	549			:8 RESERVED
				:4 JOB HAS USER DATA AREA
				:2 CURRENT PROCESS REQUIRES USER DATA AREA
				:1 USER DATA AREA IN MEMORY
MIX-TF	550	1	N	:8 PROCESS TO BE PREEMPTED
				:4 RESERVED
				:2 DATA TO BE TRANSFERRED FROM HANDLER TO UDA
				:1 FULL TIME-SLICE HAS BEEN UTILIZED
	551	1	N	RESERVED
MIX-TS	552-555	4	N	TIME-SLICE FOR TIME SHARING PROCESS
MIX-D1	556-571	16	N	DISK ADDRESS OF SWAP-OUT CODE FILE OF CURRENT PROCESS
MIX-D2	572-587	16	N	DISK ADDRESS OF USER DATA AREA STORAGE
MIX-BP	588-589	2	N	PIB INDEX OF BASE PROCESS
	590-599	10	N	RESERVED

```
          MAINTENANCE LOG RECORD                                  ML-===

          The following is a description of the Maintenance Log file.

******************************************************************
*        MAINTENANCE LOG RECORD * MAINTENANCE LOG RECORD         *
******************************************************************

          Common to all Maintenance Log Records

ML-TYP       0    1 N     MAINTENANCE LOG RECORD TYPE CODE
                          0 = MAINTENANCE LOG HEADER RECORD
                                     (LOG HEAD, H/L, LOG TAIL,
                                      CONFIGURATION)
                          1 = DEVICE OPEN RECORD
                          2 = DEVICE CLOSE RECORD
                          3 = DEVICE FAILURE RECORD
                          4 = MEMORY FAILURE RECORD
                          5 = MAINTENANCE LOG COMMENT RECORD
                          6 = SPURIOUS I/O COMPLETE RECORD,
                                 PROCESSOR ERROR, SHORT BMDUMP
                          7 = MS-2 NORMAL STATE SNAP PICTURE,
                                 FULL BMDUMP
                          8 = MAINTENANCE LOG TRANSFER RECD,
                                 CONFIGURATION CHANGES
                          9 = RESERVED
          1- 4    4 N     RESERVED
ML-RLG    5- 8    4 N     RUN LOG ID NUMBER
ML-DAT    9-14    6 N     MAINTENANCE LOG RECORD DATE STAMP
ML-TIM   15-22    8 N     MAINTENANCE LOG RECORD TIME STAMP
ML-SUB      23    1 N     MAINTENANCE LOG RECORD SUB-TYPE CODE
         24-99   76 N     RECORD BODY
```

```
          MAINTENANCE LOG RECORD                              ML-===
                                                                (Continued)

*********************************************************************
*
*      TYPE 0/0, 0/1, 0/2   MAINTENANCE LOG HEADER RECORD
*
*                SUB-TYPE 0 = HEADER RECORD
*                SUB-TYPE 1 = HALT / LOAD RECORD
*                SUB-TYPE 2 = TRAILER RECORD
*

 ML-MCP   24-35   6 A    MCP ID
 ML-NAM   36-69  17 A    PROCESSOR NAME
 ML-ASR   70-77   4 A    MCP ASR #
 ML-VSN   78-83   6 N    MCP VERSION DATE
 ML-KD    84-87   4 N    PROCESSOR MEMORY SIZE IN K DIGITS
 ML-PR#      88   1 N    PROCESSOR NUMBER
 ML-CPU      89   1 N    PROCESSOR TYPE CODE
 ML-MEM      90   1 N    PROCESSOR MEMORY TYPE CODE
 ML-POP      91   1 N    PROCESSOR OPTIONS
                         :8 MS-2 PROCESSOR
                         :4 EXTENDED ADDRESSING
                         :2 ACCUMULATOR
                         :1 FLOATING POINT
 ML-HLC      92   1 N    HALT LOAD CAUSE
 ML-LOD      93   1 N    LOADER FLAG
          94-99   6 N    RESERVED

*
*********************************************************************
*
*      TYPE 0/5               MAINTENANCE LOG HEADER RECORD
*
*                SUB-TYPE 5 = SYSTEM CONFIGURATION RECORD
*

 ML-STA  24-25    2 N    DEVICE IOAT POINTER (IO-STA)
 ML-HDW  26-27    2 N    DEVICE HARDWARE TYPE
 ML-HDS     28    1 N    DEVICE HARDWARE SUBTYPE CODE
 ML-UNT     29    1 N    DEVICE UNIT NUMBER
 ML-CHN  30-31    2 N    DEVICE CHANNEL NUMBER
 ML-MID  32-43    6 A    MULTI FILE ID
         44-83   40 N    <<REPEAT 2 ENTRIES AS 24-43 ABOVE>>
         84-99   16 N    RESERVED
```

MAINTENANCE LOG RECORD ML-===
(Continued)

```
*
**********************************************************************
*
*     TYPE 1/0                    DEVICE OPEN RECORD
*

 ML-STA  24-25    2 N    DEVICE IOAT POINTER (IO-STA)
 ML-HDW  26-27    2 N    DEVICE HARDWARE TYPE
 ML-HDS     28    1 N    DEVICE HARDWARE SUBTYPE CODE
 ML-UNT     29    1 N    DEVICE UNIT NUMBER
 ML-CHN  30-31    2 N    DEVICE CHANNEL NUMBER
 ML-PID  32-43    6 A    PROGRAM ID
 ML-PMF  44-55    6 A    PROGRAM MULTI ID
 ML-MIX  56-57    2 N    PROGRAM MIX NUMBER
 ML-FID  58-69    6 A    FILE ID
 ML-MFD  70-81    6 A    MULTI FILE ID
 ML-RL#  82-84    3 N    MAGNETIC TAPE REEL NUMBER
            85    1 N    RESERVED
 ML-SR#  86-91    6 N    MAGNETIC TAPE PHYSICAL CAN NUMBER
                         DISK PACK SERIAL NUMBER
 ML-BKO  92-99    8 N    DEVICE PHYSICAL BLOCK COUNT

*
**********************************************************************
*
*     TYPE 2/0                    DEVICE CLOSE RECORD
*
 ML-STA  24-25    2 N    DEVICE IOAT POINTER (IO-STA)
 ML-HDW  26-27    2 N    DEVICE HARDWARE TYPE
 ML-HDS     28    1 N    DEVICE HARDWARE SUBTYPE CODE
 ML-UNT     29    1 N    DEVICE UNIT NUMBER
 ML-CHN  30-31    2 N    DEVICE CHANNEL NUMBER
 ML-ERC  32-35    4 N    TOTAL ERROR COUNT ON DEVICE
 ML-RTC  36-37    2 N    RETRY COUNT SPECIFIED FOR DEVICE
 ML-BKC  38-45    8 N    TOTAL BLOCK COUNT ON DEVICE
         46-99   54 N    RESERVED
```

MAINTENANCE LOG RECORD ML-===
(Continued)

```
*
********************************************************************
*
*     TYPE 3/0                 DEVICE FAILURE RECORD
*
 ML-STA  24-25     2 N    DEVICE IOAT POINTER (IO-STA)
 ML-HDW  26-27     2 N    DEVICE HARDWARE TYPE
 ML-HDS     28     1 N    DEVICE HARDWARE SUBTYPE CODE
 ML-UNT     29     1 N    DEVICE UNIT NUMBER
 ML-CHN  30-31     2 N    DEVICE CHANNEL NUMBER
 ML-BAS  32-35     4 N    BASE ADDRESS IN KD FOR I/O DESCRIPTOR
 ML-DSC  36-59    24 N    DEVICE I/O DESCRIPTOR
 ML-OCH  60-61     2 N    ORIGINAL CHANNEL HANDLING I/O
 ML-RCH  62-63     2 N    CHANNEL ON WHICH I/O RECOVERED
 ML-RTY  64-65     2 N    NUMBER OF RETRIES ATTEMPTED
 ML-ESZ  66-67     2 N    MAGNETIC TAPE ERASE SIZE IN 100 DIGITs
 ML-BLK  68-75     8 N    BLOCK COUNT AT TIME OF ERROR
 ML-FIX  76-77     2 N    ERROR FIX CODE
 ML-RCV     78     1 N    DEVICE RECOVERY STATUS
                                 0 = GOOD I/O ERROR RECOVERY
                                 1 = NO RECOVERY - NEW ERROR
                                 2 = NO RECOVERY - RETRIES THRU
 ML-ST3     79     1 N    LCP TYPE DEVICE FLAG
                :4        DEVICE IS LCP TYPE DEVICE
 ML-R/D  80-95    16 N    ORIGINAL ERROR RESULT DESCRIPTOR
 ML-BIN     96     1 N    BINARY ADDRESSING FLAG
         97-99     3 N    RESERVED

*
********************************************************************
*
*     TYPE 4/0, 4/1, 4/2   MEMORY FAILURE RECORD
*
*               SUB-TYPE 0 = NORMAL MEMORY ERROR REPORT
*               SUB-TYPE 1 = ERROR REPORTING SUSPENDED DUE
*                            TO EXCESSIVE ERRORS
*               SUB-TYPE 2 = ERROR REPORTING RESUMED
*
 ML-HAM  24- 25     2 N   HAMMING CODE
 ML-ADH  26- 29     4 N   HIGH ORDER PART OF FAILURE ADDRESS
 ML-ADL  30- 33     4 N   LOW ORDER PART OF FAILURE ADDRESS
 ML-RSL  34- 35     2 N   MEMORY ERROR RESULT
 ML-CNT  36- 39     4 N   ERROR COUNT AT THIS ADDRESS
         40-199   160 N   REPEAT 10 ENTRIES AS 24-39 ABOVE
```

MAINTENANCE LOG RECORD ML-===
(Continued)

```
*
***********************************************************************
*
*     TYPE 5/0              MAINTENANCE LOG COMMENT RECORD
*
*                 NOTE THAT TYPE 5 RECORDS ARE 200 DIGITS LONG
*
 ML-COM   24-173 75 A    MAINTENANCE LOG COMMENT FIELD
         174-199 26 N    RESERVED

*
***********************************************************************
*
*     TYPE 6/0              SPURIOUS I/O COMPLETE RECORD
*
         24-59   36 N    RESERVED
 ML-OCH  60-61    2 N    SPURIOUS CHANNEL NUMBER
         62-79   18 N    RESERVED
 ML-R/D  80-95   16 N    SPURIOUS RESULT DESCRIPTOR
         96-99    4 N    RESERVED

*
***********************************************************************
*
*     TYPE 6/1,6/2,6/3,6/4  PROCESSOR ERROR RECORD
*
*                SUB-TYPE 1 = OVER TEMPERATURE WARNING RECORD
*                SUB-TYPE 2 = TEMPERATURE BACK TO NORMAL RECORD
*                SUB-TYPE 3 = UNDER VOLTAGE WARNING RECORD
*                SUB-TYPE 4 = VOLTAGE BACK TO NORMAL RECORD
*
         24-99   76 N    RESERVED

*
***********************************************************************
*
*     TYPE 6/5               SHORT BUFFER MEMORY DUMP RECORD
*
 ML-STA  24-25    2 N    DEVICE IOAT POINTER (IO-STA)
 ML-HDW  26-27    2 N    DEVICE HARDWARE TYPE
 ML-HDS     28    1 N    DEVICE HARDWARE SUBTYPE CODE
 ML-UNT     29    1 N    DEVICE UNIT NUMBER
 ML-CHN  30-31    2 N    DEVICE CHANNEL NUMBER
 ML-BAS  32-35    4 N    BASE ADDRESS IN KD FOR I/O DESCRIPTOR
 ML-DSC  36-59   24 N    DEVICE I/O DESCRIPTOR
 ML-BMS  60-99   40 N    BUFFER MEMORY DUMP
```

```
            MAINTENANCE LOG RECORD                              ML-===
                                                                   (Continued)

      *
      *******************************************************************
      *
      *     TYPE 6/6                 HUNG DLP STATUS DUMP RECORD
      *
       ML-STA   24- 25    2 N     DEVICE IOAT POINTER (10-STA)
       ML-HDW   26- 27    2 N     DEVICE HARDWARE TYPE
       ML-HDS       28    1 N     DEVICE HARDWARE SUBTYPE CODE
       ML-UNT       29    1 N     DEVICE UNIT NUMBER
       ML-CHN   30- 31    2 N     DEVICE CHANNEL NUMBER
       ML-BAS   32- 35    4 N     BASE ADDRESS IN KD FOR I/O DESCRIPTOR
       ML-DSC   36- 59   24 N     DEVICE I/O DESCRIPTOR
       ML-BMS   60- 99   40 N     DLP STATUS DUMP
                96- 99    4 N     RESERVED

      *
      *******************************************************************
      *
      *     TYPE 7/0                 MS-2 NORMAL STATE SNAP PICTURE
      *
      *                    NOTE THAT TYPE 7 RECORDS ARE 400 DIGITS LONG.
      *
       ML-SP#   24- 25    2 N     SNAP PICTURE SECTION NUMBER
                26- 31    6 N     RESERVED
       ML-PID   32- 43    6 A     PROGRAM ID
       ML-PMF   44- 55    6 A     PROGRAM MULTI ID
       ML-MIX   56- 57    2 N     PROGRAM MIX NUMBER
                58- 59    2 N     RESERVED
       ML-SNP   60-399  340 N     SNAP PICTURE DATA

      *
      *******************************************************************
      *
      *     TYPE 7/1                 FULL BUFFER MEMORY DUMP RECORD
      *
      *                    NOTE THAT TYPE 7 RECORDS ARE 400 DIGITS LONG.
      *
       ML-STA    24- 25    2 N    DEVICE IOAT POINTER (IO-STA)
       ML-HDW    26- 27    2 N    DEVICE HARDWARE TYPE
       ML-HDS        28    1 N    DEVICE HARDWARE SUBTYPE CODE
       ML-UNT        29    1 N    DEVICE UNIT NUMBER
       ML-CHN    30- 31    2 N    DEVICE CHANNEL NUMBER
       ML-BAS    32- 35    4 N    BASE ADDRESS IN KD FOR I/O DESCRIPTOR
       ML-DSC    36- 59   24 N    DEVICE I/O DESCRIPTOR
       ML-BMD    60- 67    8 N    FIRST 2 WDS OF R/D FROM BMDUMP OP INIT
       ML-BMS    68-371  304 N    BUFFER MEMORY DUMP
                372-399   28 N    RESERVED
```

```
        MAINTENANCE LOG RECORD                              ML-===
                                                              (Continued)

*
**********************************************************************
*
*  TYPE 8/0, 8/1, 8/2        MAINTENANCE LOG TRANSFER RECORD
*
*                    SUB-TYPE 0 = DUMMY OPEN RECORD (SEE TYPE 1/0)
*                    SUB-TYPE 1 = DUMMY CLOSE RECORD (SEE TYPE 2/0)
*                    SUB-TYPE 2 = PACK MOUNT RECORD (SEE TYPE 1/0)

*
**********************************************************************
*
*     TYPE 8/3                UNIT ADD RECORD
*
 ML-STA  24-25    2 N     DEVICE IOAT POINTER (IO-STA)
 ML-HDW  26-27    2 N     DEVICE HARDWARE TYPE
 ML-HDS     28    1 N     DEVICE HARDWARE SUBTYPE CODE
 ML-UNT     29    1 N     DEVICE UNIT NUMBER
 ML-CHN  30-31    2 N     DEVICE CHANNEL NUMBER
         32-69   38 N     RESERVED
 ML-MFD  70-81    6 A     MULTI FILE ID
         82-99   18 N     RESERVED

*
**********************************************************************
*
*     TYPE 8/4, 8/5, 8/6     UNIT NOT AVAILABLE RECORD
*
*                    SUB-TYPE 4 = UNIT DELETE RECORD
*                    SUB-TYPE 5 = UR RECORD
*                    SUB-TYPE 6 = DBMM RECORD
*
 ML-STA  24-25    2 N     DEVICE IOAT POINTER (IO-STA)
 ML-HDW  26-27    2 N     DEVICE HARDWARE TYPE
 ML-HDS     28    1 N     DEVICE HARDWARE SUBTYPE CODE
 ML-UNT     29    1 N     DEVICE UNIT NUMBER
 ML-CHN  30-31    2 N     DEVICE CHANNEL NUMBER
 ML-POM  32-33    1 A     TYPE "+" OR "-" (USED FOR 8/5 AND 8/6)
         34-99   66 N     RESERVED
```

```
          MAINTENANCE LOG RECORD                              ML-===
                                                                  (Continued)

*
***********************************************************************
*
*      TYPE 8/7                  XC RECORD
*
              24-29     6 N     RESERVED
  ML-CHN      30-31     2 N     DEVICE CHANNEL RECORD
  ML-POM      32-33     1 A     TYPE "+" OR "-" SUBTYPE CODE
              34-99    66 N     RESERVED

*
***********************************************************************
*
*      TYPE 8/8                  MEMORY, DISK, PACK RECORD
*
*                       CODE 0 = WXM (USED FOR SYSTEM CONFIGURATION)
*                       CODE 1 = WXD (USED FOR SYSTEM CONFIGURATION)
*                       CODE 2 = WXP (USED FOR SYSTEM CONFIGURATION)
*                       CODE 3 = XM
*                       CODE 4 = XD
*                       CODE 5 = XP
*                       CODE 6 = RXM
*                       CODE 7 = RXD
*                       CODE 8 = RXP
*
  ML-COD         24     1 N     CODE
                 25     1 N     RESERVED
  ML-EU#      26-27     2 N     EU# (NOT USED FOR CODES 0, 3, OR 6;
                                       EU# FOR DISK< IO-STA VALUE FOR PACK)
  ML-ADR      28-37    10 N     ADDRESS
  ML-SIZ      38-47    10 N     SIZE (NOT USED FOR CODES 7 OR 8)
              48-91    44 N     REPEAT 2 ENTRIES AS 26-47 ABOVE
                                IF CODES 0, 1, OR 2
              92-99     8 N     RESERVED

*
***********************************************************************
*
*         TYPE 9/0               FILLER ENTRY
*
              24-99    76 N     RESERVED
```

MCS QUEUE MQ-===

The MCS Queue is allocated only if specifically requested by the MCS "ATTACH MCS QUEUE" function.

MQ-MCS	0- 7	8	N	POINTER TO MCS TABLE ENTRY
MQ-ID#	8- 9	2	N	MCS ID NUMBER
MQ-SIZ	10- 15	6	N	QUEUE ENTRY SIZE IN DIGITS
MQ-AVC	16- 19	4	N	NUMBER OF AVAILABLE ENTRIES
MQ-USC	20- 23	4	N	NUMBER OF ENTRIES IN USE
MQ-AVL	24- 31	8	N	FIRST AVAILABLE ENTRY
MQ-NXT	32- 39	8	N	NEXT TO PROCESS
MQ-EOQ	40- 47	8	N	END OF Q POINTER
MQ-QBS	48- 51	4	N	QUEUE ENTRY TEXT SIZE
	52- 99	48	N	RESERVED
MQ-HDR	100-179	80	N	HEADER
MQ-TXT	180-			TEXT IF ANY

MASTER AVAILABLE DISK TABLE MST-==

The Master Available Disk Table is 2000 digits long (10 segments). All disk assigned to the system (all processors) is contained in this table. The disk address of this Master Available Disk Table is found in the Halt/Load parameters. Of the 125 entries in the table, the first entry is a dummy containing all 9s. The last entry is a dummy containing all zeros. The dummy entries eliminate endpoint problems. The remaining 123 entries are initially set to all 9s. Available disk entries are kept in descending sequence at the top of the table.

MST-AV	0-199		MASTER AVAILABLE DISK TABLE
MST-EL	0- 116	N	ENTRY #1
MST-SS	0- 9	N	ENTRY #1: AVAILABLE DISK ADDRESS
MST-EU	0- 1	N	ENTRY #1: EU NUMBER OF DISK ADDRESS
MST-AD	2- 7	N	ENTRY #1: REMAINDER OF DISK ADDRESS
MST-LN	9- 17	N	ENTRY #1: NUMBER OF AVAILABLE SEGMENTS
	16-1999		ENTRIES #2 THRU #125

STATUS NS-===

NS-TIM THRU NS-TED: TABLE OF PERIODIC TIME ROUTINES **

Each entry is 20 digits long and can be used to initiate the execution of a global routine or overlay process at a specific time or after a specific number of seconds have elapsed. Table evaluation occurs every n-second period (usually every second). Uninterruptable (global) routines occur first in the table, followed by the overlayable routines. Routines with a wake-up time of 99999 are non-functional. A routine may have a wake-up time or a wake-up period, but generally should not have both. When a routine is scheduled for execution it is inhibited, and is not uninhibited until it completes. If an inhibited routine is detected, no further attempt is made to execute any periodic function until the inhibit is removed. In this manner it is possible to execute the global routines even though an overlay routine may still be running, however an overlay routine will not be scheduled while a previous copy of itself is still running.

Format of the table:

0- 4	5 N	NEXT TIME ROUTINE IS TO BE CALLED IN SECONDS (99999 = NONFUNCTIONAL).
5- 9	5 N	FREQUENCY OF EXECUTION IN SECONDS MINUS ONE (EX: 00009 = EX EVERY 10 SECONDS). (8 BIT OF FIRST DIGIT IS USED AS INHIBIT FLAG).
10- 11	2 N	SGNM OF OVERLAY TO CALL (00 FOR GLOBAL).
12- 14	3 N	00 FOR OVERLAY CALL, AVAILABLE FOR GLOBAL.
15- 19	5 N	ACON OF ROUTINE TO EXECUTE.

Status Device Test Initiate Routine (NSTEST)

NS-HDW TABLE - 2-DIGIT HARDWARE TYPE,
1-DIGIT POSITION RELATIVE TO START OF OP OF STANDARD UNIT NUMBER
1-DIGIT POSITION RELATIVE TO START OF OP OF LCP UNIT NUMBER
1-DIGIT NEW IO-NSC VALUE IF IO-NSC = 9
1-DIGIT VALUE OF FIRST VARIANT IN LCP TEST OP

OPEN/CLOSE BLOCK LABEL GROUP HEADER OCG===

The OPEN/CLOSE Block Group Header (OCGHDR) begins every label group (BOV, BOF, EOV, EOF). It starts on a segment boundry.

OCGHDR	0- 3		OPEN/CLOSE BLOCK LABEL GROUP HEADER
OCG#TM	0	1 N	NUMBER OF TAPEMARKS AFTER LABEL GROUP
OCG#EX	1	1 N	NUMBER OF EXTRA TAPEMARKS AT EOT
OCG#RD	2- 3	2 N	NUMBER OF OPEN/CLOSE RECORD DESCRIPTIONS

OPEN/CLOSE BLOCK LABEL RECORD HEADER OCR===

There is one OPEN/CLOSE Block Record Header (OCDB) for every label type which may appear in a label group. The OCDB serves as a header for the OPEN/CLOSE field descriptions which follow it.

OCRD	0-17		OPEN/CLOSE BLOCK RECORD DESCRIPTION
OCROCR	0- 3	4 N	NEXT OCRD LINK
OCRMIN	4- 5	2 N	MINIMUM# OF TIMES USED (0= OPTIONAL)
OCRMAX	6- 7	2 N	MAXIMUM# OF TIMES USED(@FF@ = NO LIMIT)
OCR-AC	8	1 N	ACCESS FLAG
OCR-AC			=0 MCP ONLY MAY ACCESS
OCR-AC			=1 USER MAY READ BUT NOT WRITE
OCR-AC			=3 USER MAY READ OR WRITE
OCR-AC			>=8 ONLY VALID IF MATCHING SYSCODE
	9	1 N	FILLER
OCR#RF	10-11	2 N	NUMBER OF FIELD DESCRIPTIONS
OCRMRL	12-15	4 N	MINIMUM LENGTH FOR LABEL RECORD
OCRSTR	16-17	2 N	FILLER STRING VALUE (=@40@)

DCP OUTPUT HEADER BUFFER OH-===

OH-DFR	0- 79	80	N	DEFAULT READ I/O BUFFER
	80- 99	20	N	RESERVED
OH-STA	100-179	80	N	MCS FILE ENABLE STATUS HEADER ARRAY ONE DIGIT PER MCS INDEXED BY MCS NUMBER FOR MCS STATUS CHANGE (FUNCTION 66) TO DCP 0 = DCP INPUT FOR MCS DIS-ENABLED, SET WHEN RESULT HEADER QUE IS FULL TO STOP INPUT 1 = DCP INPUT FOR MCS ENABLED
OH-MDP	180-183	4	N	FFFF OR PTR TO S MEMORY BUFFER TO BE DEALLOCATED. FFFF = NOTHING TO DEALLOCATE
OH-FST	184-191	8	N	ABS ADR FIRST MCS TABLE ENTRY WTG DCP IIO
OH-LST	192-199	8	N	ABS ADR LAST MCS TABLE ENTRY WTG DCP IIO
OH-IOA	200-207	8	N	ABS ADR HARD IOAT FOR THE DCP
OH-SDB	208-223	16	N	THIS DCPS CODEFILE BEGINING DISK ADDRESS
OH-SDE	224-239	16	N	THIS DCPS CODEFILE ENDING DISK ADDRESS
OH-LVP	240-243	4	N	HEX ADDRESS OF DCP LINE VECTOR TABLE IN S MEMORY
OH-ACK	244-245	2	N	DEFAULT READ ACK/NAK FLAG sent to the DCP as part of the default read I/O descripter to acknowledge the good receipt of a result header and there was space to store it in the MCS's RESULT QUEUE. 00 = ACK, FF = NAK
OH-DC#	246	1	N	DCP TABLE INDEX FOR THIS DCP (INDEX OF THIS ENTRY)
OH-DVF	247	1	N	DCP I/O-IN-PROGRESS FLAG (F=TRUE) EXCEPT DEFAULT READ
OH-MCN	248	1	N	STATUS CHANGE I/O REQUESTED (F=TRUE)
OH-RYN	249	1	N	MAKE DCP READY I/O REQUESTED (F=TRUE)
OH-TSN	250	1	N	TEST OP I/O REQUESTED (F=TRUE)
	251-299	49	N	RESERVED

OBJECT PROGRAMS ON DISK

When a program is compiled, the generator creates an executable object program on disk. The format of that code file is fixed so the MCP can easily determine the necessary program requirements and load the program into memory. The code file has the following components:

Program Parameter Block

File Parameter Block(s)

Object Program

The code file occupies one disk area, no greater than 9999 segments and subject to the restrictions noted in the following sub-sections. Compilers generate only one-area code files.

OBJECT PROGRAMS ON DISK (Continued)

The Program Parameter Block (PPB) of the code file contains information needed to schedule and load the object program and always occurs at the beginning (zero-th segment) of the code file. The PPB contains two basic groups of data; general program information and the program Segment Dictionary. The former is used primarily to schedule and load the program into memory; the latter is primarily used during the load procedure to build the program Segment Dictionary in memory.

Aside from the Segment Dictionary, the fields of the PPB do not reside in memory.

OBJECT PROGRAMS ON DISK (Continued)

The layout of the Program Parameter Block follows:

Initial Block of Program Parameter Block

0- 31	16	A	PPB Identification
32- 63	32	N	Program Segment Dictionary Master Entry
64- 95	32	N	Segment 1
96-127	32	N	Segment 2
128-159	32	N	Segment 3
160-191	32	N	Segment 4
192-199	8	N	PPB Initial Block Identification

Second (and all successive) Block of Program Parameter Block

0- 31	32	N	Segment 5	(11...)
32- 63	32	N	Segment 6	(12...)
64- 95	32	N	Segment 7	(13...)
96-127	32	N	Segment 8	(14...)
128-159	32	N	Segment 9	(15...)
160-191	32	N	Segment 10	(16...)
192-199	8	N	Reserved	

The following information is the PPB identification.

PB-PRN	0-11	6	A	Program name
PB-SGS	12-14	3	N	Number of overlayable segments
PB-INS	15-19	5	N	Address of first executable instruction
PB-COR	20-25	6	N	Memory requirements
PB-SDA	26-31	6	N	Relative memory address of Segment Dictionary

The following information is the Program Segment Dictionary master entry.

PB-BCT	32- 37	6	N	BCT instruction 300174
PB-DFD	38- 39	2	N	Number of disk files declared
PB-FPF	40	1	N	File Parameter Block Flag 0 = NO FPBs 1 = 100 digits (usual size) 2,3 = 200 digits 4,5 = 200 digits + Port Blocks
PB-OPS	41- 43	3	N	Number of logical segments
PB-WFL	44- 49	6	N	Workflow Language flag (="WFL")
PB-OVN	50- 52	3	N	Segment number requested for overlay
PB-BSG	53- 55	3	N	001
PB-CDT	56- 61	6	N	Date program compiled
PB-CPL	62- 63	1	A	Compiler of this Program

OBJECT PROGRAMS ON DISK (Continued)

A	=	ASSMBLR
B	=	BPL/BINDER/BASIC
C	=	COBOL
D	=	DASDL
F	=	FORTRAN
P	=	PASCAL
R	=	RPG
W	=	WFL
X	=	XFORTN

OBJECT PROGRAMS ON DISK (Continued)

The following are Program Segment Dictionary working entries.

PB-PRB	0- 5	6 N	Address of PB-BCT or first instruction of segment
PB-RDA	6- 11	6 N	Disk address of program segment relative to beginning of code file
PB-SLO	12- 17	6 N	Address of first instruction to be executed after overlay call
PB-BEG	18- 23	6 N	Beginning memory address of overlay
PB-END	24- 29	6 N	Ending memory address of overlay
PB-LVL	30- 31	2 N	After program load, EU and high order digit of absolute disk address of program

The working entries are repeated for each overlay in the program using the first 192 digits of additional disk segments, if necessary.

The following two fields are found only in the first disk segment of Program Parameter Block.

PB-FIL	192-193	2 N	Number of files declared in program
PB-MSZ	194-199	6 N	Memory size of global segment (digits)

The following data is used during program scheduling:

PB-BCT - Must contain 300174 (branch communicate instruction). Identifies file as a program.

PB-COR - Memory required for program.

Program Loading

The following fields are used during the loading of the first segment of the program into memory:

PB-SGS - To calculate Segment Dictionary size.

PB-SDA - Address where Segment Dictionary to be built.

The first 23 digits of the master Segment Dictionary entry and all working Segment Dictionary entries are placed into memory during program loading.

The following fields are used to skip past the FPBs and find the beginning of the non-overlayable segment of the program code.

OBJECT PROGRAMS ON DISK (Continued)

PB-FPF - Flag defining size of indvidual file parameter blocks (FPB).

PB-FIL - Number of FPBs.

The following fields are used to load the program and begin its execution:

PB-INS - Relative memory address of first program instruction.

PB-MSZ - Size of non-overlayable segment is used to construct disk read descriptor to load program.

Miscellaneous

The following fields are used by the DC console message:

PB-BCT - As above.

PB-PRN - Program name as defined in source program, or the time of the compile.

PB-COR, PB-DFD - Memory required.

PB-CDT - Date compiled.

PB-CPL - First character of compiler name.

The remainder of the PPB fields constitute the Segment Dictionary which is used for program overlay requests.

Immediately following the PPB in an object program on disk is a full disk segment.

OBJECT PROGRAMS ON DISK (Continued)

Program Code

Following the FPBs in a code file is the object program's non-overlayable segment and then the program overlays. All program segments begin on disk segment boundaries; thus a small amount of unused space can exist between program segments.

The program code file relative addresses of the overlayable segments can be found in the individual segment dictionary entries (PB-RDA). (All addresses are zero relative.)

DISK PACK FILE HEADER (ON DISK PACK) PF-===

The following descriptions apply to the disk pack version of the Disk File Header.

PF-CAD	0- 5	6	N	Zeros
PF-SPT	6- 13	8	N	Header size in bytes
PF-SIZ	14- 17	4	N	Header size in bytes
PF-TP1	18	1	N	File type
				:8 Reserved
				:4 File name change in progress
				:2 Incomplete file (partially removed)
				:1 Split cylinder file
PF-TP2	19	1	N	File type
				:8 Assign by space available file
				:4 Assign by area file
				:2 Single pack
				:2/ Multipack file
				:1 Cylinder bound file
PF-TP3	20	1	N	File type
				:8 Reserved
				:4 SET = Inhibit APCR and APBD options for this file. "RN=", "PBP=", and "PCP=" commands inhibited for this file.
				:2 Reserved
				:1 No squash file
PF-ORG	21	1	N	
				:8 Reserved
				:4 Indexed I/O Key File
				:2 Indexed I/O Data File
				:1 Relative I/O Data File
	22- 23	2	N	Not used
PF-BEN	24- 31	8	N	Block EOF pointer
PF-RSZ	32- 37	6	N	Record size in digits
PF-RPB	38- 40	3	N	Records per block
PF-BSZ	41- 49	9	N	Block size in digits
PF-BPA	50- 51	6	N	Blocks per area
PF-SPA	56- 61	6	N	Sectors per area
PF-#AR	62- 65	4	N	Areas requested
PF-UAR	66- 69	4	N	Area counter
PF-EOF	70- 79	10	N	EOF pointer
PF-FRM	80- 81	2	N	Record format
	82- 89	8	N	Reserved
PF-PPA	90- 93	4	N	Partitions per area (split cylinder files)
PF-CDT	94- 98	5	N	Creation date
PF-LAD	99-103	5	N	Last access date
PF-SAV	104-108	5	N	Save factor
PF-ADB	109-116	8	N	Base pack header address
PF-BSN	117-122	6	N	Base pack serial number
	123-125	3	N	Reserved

DISK PACK FILE HEADER (ON DISK PACK) PF-===
(Continued)

PF-SNS	126	1	N	Sensitivedata flag
PF-STY	127	1	N	Security type
				8 = Controlled
				4 = Guarded
				2 = Public
				1 = Private
				0 = None
PF-SUS	128	1	N	Security use
				6 = IO (default)
				4 = IN
				2 = OUT
				1 = SECURED
PF-MA1	129	1	N	Security Maint (not implemented)
				:8 ADD
				:4 DUMP
				:2 CHANGE
				:1 REMOVE
PF-SUC	130-149	20	N	Usercode
PF-GRD	150-161	12	N	Guard file ID
PF-OTY	162	1	N	Open type
PF-PRM	163	1	N	Permanent flag
PF-NU1	164-165	1	N	Number of users processor 1
PF-OO1	166-167	2	N	Number of open out processor 1
PF-NU2	168-169	2	N	Number of users processor 2
PF-OO2	170-171	2	N	Number of open out processor 2
PF-NU3	172-173	2	N	Number of users processor 3
PF-OO3	174-175	2	N	Number of open out processor 3
PF-NU4	176-177	2	N	Number of users processor 4
PF-OO4	178-179	2	N	Number of open out processor 4
PF-BPP	180-183	4	N	Blocks per partition
PF-FAM	184-195	12	N	Pack ID of the guard file
PF-MIX	196-197	2	N	Mix number of generator or locking program
PF-SYS	198	1	N	Number of generator or locking system
PF-NRM	199	1	N	
				:8 Remove this file on null activity
PF-AR1	200-207	8	N	First area link
	208-359	152	N	19 more area links

PORT FILE INTERFACE BLOCK PFB===

Also known as the "PORT FIB" or "PFIB". This is the User Program Interface to Port Files. It resides in the user's program and is used to return useful data to the user after each I/O request, and to pass information to the MCP which usually changes with each request (e.g. the actual key value).

PORTFB	0-199			PORT FILE INTERFACE BLOCK
PFBVA	0- 3	4	N	FIB VALIDATION FLAG = @CACA@
PFBLEV	4- 5	2	N	FIB REVISION LEVEL = 01
PFBIOA	6- 11	6	N	EXTERNAL PORT BLOCK LINK
PFBFNM	12- 14	3	N	RELATIVE FILE NUMBER DECLARED BY COMPILER
	15- 43	29	N	<<AVAILABLE>>
PFBBSZ	44- 49	6	N	BUFFER SIZE IN BYTES
PFBBUF	50- 55	6	N	BUFFER ADDRESS (MUST BE MOD4)
	56- 79	24	N	<<AVAILABLE>>

OUTPUT PARAMETERS (FUNCTION RETURN VALUES)

PFBSUB	80- 83	4	N	SUBPORT INDEX OF LAST OPERATION
PFBMSZ	84- 89	6	N	MAX-MESSAGE-TEXT-SIZE IN BYTES
PFBERR	90- 91	2	N	ERROR VALUE FROM PRECEEDING OPERATION 0 = NOERROR 1 = DISCONNECTED 2 = DATALOST (ON CLOSE) 3 = NOBUFFER (ON WRITE) 4 = NOFILEFOUND (ON OPEN AVAILABLE) 5 = UNREACHABLEHOST (ON OPEN)
PFBSTA	92	1	N	CURRENT SUBPORT STATE 1 = CLOSED 2 = OPEN-PENDING 3 = OPENED 4 = BLOCKED 5 = AWAITING-HOST 6 = DEACTIVATED 7 = CLOSE-PENDING 8 = CLOSE-BLOCKED 9 = DEACTIVATION-PENDING A = ALMOST-OPENED B = SHUTDOWN-IN-PROCESS C = NEVER-OPENED
PFBEOF	93	1	N	EOF FLAG 0 = NO EOF ON LAST OPERATION 1 = EOF DETECTED ON LAST OPERATION
PFBMSG	94- 99	6	N	PORT INPUT MESSAGE QUEUE SIZE
PFBINQ	100-103	4	N	SUBPORT INPUT MESSAGE QUEUE SIZE
PFBCUR	104-109	6	N	CURRENTRECORD SIZE IN BYTES

PORT FILE INTERFACE BLOCK PFB===
(Continued)

PFBDA	110-114	5	N	JULIAN DATE WHEN LAST MESSAGE WAS READ
PFBTIM	115-124	10	N	TIME IN MS WHEN LAST MESSAGE WAS READ
	125-199	75	N	<<AVAILABLE>>

```
          TIME SHARING LANGUAGE PROCESSOR                     PIB-===
               INFORMATION BLOCK

                       64 N     BASE    PROCESS RECORD
PIB-ID      0- 11  6 A     PROCESSOR ID
PIB-SZ     12- 14  3 N     PROCESSOR SIZE IN KD.
PIB-DA     15- 30 16 N     ADDR OF PROC ON DISK
PIB-PA     31- 36  6 N     PROG ADDR TO RE-INSTATE
           37- 38  2 N
PIB-ST         39  1 N     PROCESSOR STATUS DIGIT
                           :8 <<AVAILABLE>>
                           :4 TYPE II PROCESSOR
                           :2 <<AVAILABLE>>
                           :1 <<AVAILABLE>>

PIB-PB     40- 55 16 N     PROG PARAM BLK DSK ADR
           56- 58  3 N
PIB-P#     59- 60  2 N     PROCESSOR NUMBER
           61- 63  3 N
```

DISK PACK AVAILABLE TABLE PKA-==

PKA-FL	0- 7	8	N	FORWARD LINK
PKA-BL	8- 15	8	N	BACKWARD LINK
PKA-SP	16- 23	8	N	ADDRESS OF THIS SECTOR
PKA-MK	24	1	N	MARKER
	25- 39	15	N	AVAILABLE
PKA-SZ	40- 47	8	N	LENGTH OF AVAILABLE AREA
PKA-AD	48- 55	8	N	ADDRESS OF AVAILABLE AREA
	56-359			19 MORE LENGTH-ADDRESS PAIRS

PACK MASTER AVAILABLE DISK TABLE PKM-==

PKM-FL	0- 7	8	N	FORWARD LINK
PKM-BL	8- 15	8	N	BACKWARD LINK
PKM-SP	16- 23	8	N	ADDRESS OF THIS SECTOR
PKM-MK	24	1	N	MARKER
	25- 39	15	N	AVAILABLE
PKM-SZ	40- 47	8	N	LENGTH OF AVAILABLE AREA
PKM-AD	48- 55	8	N	ADDRESS OF AVAILABLE AREA
	56-359			19 MORE LENGTH-ADDRESS PAIRS

PACK DISK LABEL PL-===

PL-VL1	0- 7	4	A	"VOL1"
PL-SR#	8- 19	6	A	PACK SERIAL NUMBER
PL-ACC	20- 21	1	A	RESERVED FOR INTERCHANGE PACKS
PL-PID	22- 55	17	A	PACK IDENTIFICATION
PL-SIC	56- 59	2	A	SYSTEMS/INTERCHANGE CODE
PL-COD	60- 61	1	A	PACK CODE
	62- 73	6	A	RESERVED (INTERCHANGE)
PL-OWN	74-101	14	A	OWNERS IDENTIFICATION

PL-RMF	102-105	2	A	RESTRICTED & MASTER FLAGS
				RM = RESTRICTED MASTER PACK
				RB = RESTRICTED PACK
				BM = MASTER PACK
				BB = SYSTEMS RESOURCE PACK
				WHERE B = BLANK

	106-157	6	A	RESERVED (INTERCHANGE)
PL-INT	158-159	1	A	BLANK
PL-VL2	160-167	4	A	"VOL2"
PL-IDT	168-177	5	A	INITIALIZATION DATE
PL-ISY	178-179	6	A	INITIALIZING SYSTEM
PL-DLK	190-205	8	A	DIRECTORY LINK
PL-MLK	206-221	8	A	MASTER AVAILABLE TABLE LINK
PL-ALK	222-237	8	A	AVAILABLE TABLE LINK
PL-FLG	238-239	1	A	INTEGRITY FLAG EBCDIC ZONE DIGIT ALWAYS UNDIGIT F NUMERIC FIELD BIT DEFINED: 8: IN USE BY PROCESSOR # 3 4: IN USE BY PROCESSOR # 2 2: IN USE BY PROCESSOR # 1 1: IN USE BY PROCESSOR # 0
PL-ERR	240-251	6	A	ACTUAL ERROR COUNT
PL-#XP	252-267	8	A	XD SECTOR COUNT
PL-VER	268-269	1	A	PACK LABEL AND HEADER VERSION 0 = Pre 6.7 Version 1 = 6.7 and onward Version
	270-359	45	A	RESERVED (INTERCHANGE)

PSEUDO READER DIRECTORY BLOCK PR-===

PR-FID	0-11	6 A	DATA FILE ID
PR-DK#	12-16	5 N	PSEUDO FILE DECK NUMBER
PR-RC#	17	1 N	RECORD WITHIN BLOCK
PR-BL#	18-21	4 N	BLOCK WITHIN AREA
PR-AR#	22-23	2 N	AREA WITHIN FILE
PR-FLG	24	1 N	
			:8 SAVE DECK
			:4 USRTBL ENTRY HAS BEEN MADE
			:2 PRIVILEGED PROGRAM
			:1 DISK PACK PSEUDO DECK
PR-MIX	25-26	2 N	MIX NUMBER OF USER
PR-ST1	27	1 N	
			:8 DECK IS OPEN AS DATA FILE
			:4 DECK IS RESERVED
			:2 END OF FILE SENSED
			:1 LABEL SENSED
PR-ST2	28-29	2 N	IO-STA OF DISK PACK ON WHICH DECK RESIDES

TIME SHARING PROCESS STACK PS-===

The following pointers are kept at the beginning of the Process Stack area which is immediately above the user data area.

PS-LST	00- 05	6	N	BASE ADDRESS OF LAST PROCESS STACK ENTRY
PS-TOP	06- 11	6	N	TOP ADDRESS OF LAST PROCESS STACK ENTRY

The base address of the subroutine stack is kept in Base+84 within the process and it must be MOD 4. The Process Stack entry has the following format:

PS-PRV	00- 05	6	N	BASE ADDRESS OF PREVIOUS PROCESS STACK ENTRY
PS-PAR	06- 11	6	N	PROGRAM ADDRESS OF CALLING PROCESS
PS-BAS	12- 14	3	N	BASE ADDRESS OF CALLING PROCESS
PS-LIM	15- 17	3	N	LIMIT ADDRESS OF CALLING PROCESS
PS-PA1	18- 23	6	N	PROGRAM ADDRS OF LAST CALLED TYPE II PROCESS
PS-BA1	24- 26	3	N	BASE ADDRS OF LAST CALLED TYPE II PROCESS
PS-LM1	27- 29	3	N	LIMIT ADDRS OF LAST CALLED TYPE II PROCESS
PS-PID	30- 41	6	A	ID OF CALLING PROCESS
PS-AL	42- 47	6	N	ADDRESS OF ACTION LABEL IN CALLING PROCESS
PS-INF	48-147	100	N	STATUS INFORMATION SAVED FOR CALLING PROCESS (BASE THRU BASE+100 OF TYPE II PROGRAM)
PS-TMP	148-167	20	N	MIX INFO FROM CALLING PROCESS
PS-MPD	168-179	6	A	MULTI-PROGRAM ID OF CALLING PROGRAM
PS-FH	180-215	36	N	MIX-FH, ETC. OF CALLING PROGRAM
PS-EH	216-251	36	N	MIX-EH, ETC. OF CALLING PROGRAM
PS-SSS	252-255	4	N	SIZE OF SUBROUTINE STACK IN WORDS
PS-STK	256-nnn			SUBROUTINE STACK OF PROCESS

PROGRAM SEGMENT DICTIONARY PSD===

The Program Segment Dictionary is created from information in the Program Parameter Block. This dictionary is used for all program overlay requests.

A Segment Dictionary is always at least 64 digits long. In other words it always has at least two 32-digit entries. The first 32 digits is the base, or header entry. The second and subsequent 32-digit groups correspond in sequence, to each program segment which exists.

Segment Dictionary Header Entry

The format of the Segment Dictionary header entry is as follows.

PSD-BC	0- 5	6	N	300174, overlay BCT
PSD-DF	6- 7	2	N	Number of disk files declared
PSD-FP	8	1	N	File Parameter Block flag
PSD-#S	9- 11	3	N	Number of logical segments in program
PSD-OC	12- 17	6	N	Overlay call counter
PSD-SG	18- 20	3	N	Requested logical segment number
PSD-BS	21- 23	3	N	Base logical segment number
	24- 25	2	N	reserved
PSD-SZ	26- 31	6	N	Segment Dictionary size in digits

The subsequent Segment Dictionary entries have the following format.

PSD-OV	0- 5	6	N	Address of overlay BCT or first instruction
PSD-DA	6- 11	6	N	Low order disk address of logical segment
PSD-FI	12- 17	6	N	Memory address of first instruction to execute
PSD-BE	18- 23	6	N	Beginning memory address of logical segment
PSD-EN	24- 29	6	N	Ending memory address of segment
PSD-EU	30- 31	2	N	Disk EU number of logical segment

PROGRAM OVERLAY MECHANISM

When a program needs a logical segment (in COBOL, GO TO or PERFORM to an overlayable section, in BPL, a procedure reference), the compiler generated code performs the following actions.

> The appropriate logical segment number is moved to PSD-SG and an indirect branch (BUN OP = 27) is executed to PSD-OV for the segment requested. Two results are possible.
>
> > If the segment is already in memory from a prior call, then PSD-OV for the segment contains the address of the first instruction in the segment. The indirect branch, in effect, will be a branch to the first instruction.
> >
> > If the segment is not in memory, then PSD-OV contains the address of the overlay BCT, PSD-BC. Then, the indirect branch will be to the overlay BCT instead of the first instruction. THE BCT 0174 is executed and the MCP reads the requested program segment from disk. After loading PSD-OV with PSD-FI, the program is reinstated at the PSD-FI address.

PORT BLOCK PT-===

This is the MCP's Port File Definition Block. It lives in a type 4 block and contains pointers to the user's PORT FIB and to all declared or possible subport blocks. It also contains the complete collection of port-level attributes and various counters and pointers used by the I/O routines.

PT-IOA	000-099			STANDARD IOAT FORMAT
PT-IN	100-133	17	A	INTNAME
PT-NAM	134-167	17	A	PORT NAME (TITLE)
PT-GRD	168-179	6	A	GUARDFILE ID
PT-GFM	180-191	6	A	GUARDFILE FAMILY
PT-STY	192	1	N	SECURITY TYPE
				1 = PRIVATE
				2 = PUBLIC
				4 = GUARDED
				8 = CONTROLLED
PT-SU	193	1	N	SECURITY USE
PT-BC	194-195	2	N	BCT TYPE
PT-MAX	196-199	4	N	MAX-SUBPORTS
PT-ALO	200-203	4	N	NUMBER OF ALLOCATED SUBPORTS
PT-OPN	204-207	4	N	NUMBER OF NOT "NEVER OPENED" SUBPORTS
PT-MYN	208-407	200	N	MY-NAME
PT-CHR	408	1	N	:8 PREFERED-CHARACTER-SET = EBCDIC
				:4 PREFERED-CHARACTER-SET = ASCII
				:2 ACCEPTABLE-CHARACTER-SET = EBCDIC
				:1 ACCEPTABLE-CHARACTER-SET = ASCII
PT-AL	409	1	N	:8 <<AVAILABLE>>
				:4 <<AVAILABLE>>
				:2 OPENABLE SUBPORT FOUND ON OPEN ALL RQST
				:1 "ALL SUBPORTS" FLAG FOR BCT PROCESSING
	410-415	6	N	MAX-MESSAGE-TEXT-SIZE
PT-RR	416-417	2	N	RESUME-READY-FACTOR
PT-MS	418-423	6	N	MESSAGE-QUEUE-SIZE
PT-LS	424-427	4	N	LAST-SUBPORT-USED
PT-IN	428-431	4	N	INPUT-EVENT SUBPORT COUNT
	432-435	4	N	NEXT INPUT-EVENT SUBPORT
	436-439	4	N	LAST INPUT-EVENT SUBPORT
PT-CH	440-443	4	N	CHANGED-EVENT SUBPORT COUNT
PT-CHF	444-447	4	N	NEXT CHANGED-EVENT SUBPORT
PT-CH	448-451	4	N	LAST CHANGED-EVENT SUBPORT
PT-RDY	452-455	4	N	READY-EVENT SUBPORT COUNT
PT-RYF	456-459	4	N	NEXT READY-EVENT SUBPORT
PT-RY	460-463	4	N	LAST READY-EVENT SUBPORT
PT-COB	464-471	8	N	COBOL FILE-STATUS ITEM ADDRESS
PT-BUF	472-479	8	N	BUFFER ADDRESS
PT-FL1	480	1	N	:8 FILE-STATUS VARIABLE IS USED
				:4 BUFFER ADDRESS IS CONSTANT
				:2 EXTERNAL FORMAT BUFFER (NO BLANK PADDING)
				:1 MAXIMUM BUFFER SIZE SUPPLIED FOR READ

PORT BLOCK

PT-===
(Continued)

PT-FL2	481	1	N	:8 INITIAL PORT ALLOCATION IN PROCESS
				:4 INITIAL SUBPORT ALLOCATION IN PROCESS
				:2 DATA LOST ON CLOSE OF SOME SUBPORT
				:1 FILE ATTRIBUTES ARE ON PACK (180 BYTE)
PT-FL	482	1	N	:8 MAGIC PLM PORT "BNA-PORT-PORT"
				:4 BROADCAST WRITE IN PROCESS
				:2 SET EOF AT END OF BROADCAST WRITE
				:1 ACTUAL KEY DECLARED FOR FILE
PT-FL4	483	1	N	<<AVAILABLE>>
PT-DSK	484-499	16	N	PORT FILE PARAMETER BLOCK DISK ADDRESS
PT-BAD	500-503	4	N	MEM BLOCK ADDRESS IN KD
PT-CN	504-507	4	N	COUNT OF SUBPORTS PROGRAM WAITING FOR
PT-BSZ	508-513	6	N	MAX BUFFER SIZE FOR READ IN BYTES
PT-SIZ	514-517	4	N	PORT BLOCK SIZE IN KD
PT-INQ	518-523	6	N	PORT INPUT QUEUE COUNT
PT-OTQ	524-529	6	N	PORT OUTPUT QUEUE COUNT
PT-NUM	530-533	4	N	SEQUENTIAL PORT NUMBER
PT-USR	534-567	17	A	MY USERCODE
	568-599	32	N	<<AVAILABLE>>
PT-SUB	600-607	8	N	ABSOLUTE ADDRESS OF 1ST SUBPORT BLOCK

```
          QUEUE BODY TABLE                                  Q-====

Q-LINK     0-  3    4 N   LINK TO NEXT QUEUE ELEMENT (NULL=NONE)
Q-IOAT     4- 11    8 N   IOAT ADDRESS (ABSOLUTE)
Q-BSW     12- 19    8 N   BUFFER STATUS WORD ADDRESS (ABSOLUTE)
Q-BASE    20- 23    4 N   I/O (PROGRAM) BASE ADDRESS IN KD
Q-STAT    24- 29    6 N   UNIT STATUS TEST ADDRESS (DUSTAT/EUSTAT)
Q-MIX     30- 31    2 N   MIX NUMBER
Q-WAIT        32    1 N
                          :8 RESERVED FOR INHIBIT
                          :4 PUSH INHIBIT
                          :2 LCP I/O CONVERTED
                          :1 DISK INHIBIT
Q-RQST        33    1 N
                          :8 PSEUDO I/O
                          :4 MCP GENERATED I/O
                          :2 OVERLAY REQUEST
                          :1 LINK ELEMENT AT QUEUE HEAD
Q-TYP1        34    1 N
                          :8 DISK OR DISKPACK I/O
                          :4 DATA COMM MULTI-LINE I/O
                          :2 INHIBIT I/O OVERTIME CHECK (DCOM I/O'S)
                          :1 RESERVED
Q-TYP2        35    1 N
                          :8 ALLOW DISK WRITE WITHIN MCP SEGS
                          :4 DESCRIPTOR HAS ABSOLUTE DATA ADDRESSES
                          :2 INHIBIT DATA END ADDRESS LOAD INTO BSW
                          :1 PHYSICAL RAD REQUESTED AS END ADDRESS
Q-TYP3        36    1 N
                          :8 SHARED PACK FPM OPERATION
                               USED BY ANYONE NEEDING TO DO AN FPM
                                OPERATION
                               WHO HAS NO FPM OF HIS OWN
                          :4 IGNORE IO-MSK
                          :2 INHIBIT KEYBOARD ERROR DISPLAY
                          :1 IGNORE UNRECOVERED ERROR
Q-EXIT        37    1 N   IOC EXIT LINK
                            0 = NO SPECIAL EXIT (USE MIX-IO EXIT)
                            1 = EXIT TO Q-RTRN
                            2 = MCP OVERLAY WAITING IOC EXIT
                            3 = OPERATOR CONTROL STATION.
                            4 = EXIT TO TESTER
                            5 = IGNORE PROG WTG IOC, CLEAR UPPER DESC
                            6 = PSEUDO READER BUFFER IOC (DISK OR PACK)
                            7 = DMSII IOC
                            8 = TRACE PRINT IOC EXIT
                            9 = MULTI-FILE SEARCH RECALL
                            A = SHARED PACK I/O EXIT
                            B = DMSII IOC TYPE 2
Q-MASK   38- 41     4 N   I/O ERROR IGNORE MASK
                          F000 = IGNORE ALL I/O ERRORS
          QUEUE BODY TABLE                                        Q-====
                                                                   (Continued)

                                THE FOLLOWING BIT CONFIGURATIONS DESCRIBE
                                THE I/O ERROR CASES TO BE IGNORED:
                                8000 = RESERVED
                                4000 = 020 (SHORT RECD)
                                2000 = 010 (LONG RECD)
                                1000 = 008 (END OF PAGE)
                                0800 = 200 (NOT READY)
                                0400 = 180
                                0200 = 140
                                0100 = 130
                                0080 = 120
                                0040 = 100
                                0020 = 080
                                0010 = 040
                                0008 = 004
                                0004 = 002
                                0002 = 001
                                0001 = AVAILIABLE
  Q-SPRT   42- 43     2 N       SPECIAL ROUTINE LINK
                                00 = NONE
                                06 = LOCAL SPO IOC
                                12 = DATA COMM IOC
                                18 = DCP
                                24 = OCS KX IN PROGRESS
                                30 = MICR/OCR IOC
                                36 = BINARY CARD READ IOC
                                48 = DIR I/O INHIBIT CHANNEL ON IOC
                                54 = RAD A & B ADDRESSES (DIRECT I/O)
                                60 = STATUS TEST/ENABLE IOC
                                66 = DISK PACK IOC
                                78 = HPT SHARED DISK LINK
                                84 = RESERVED
                                90 = RESERVED
                                96 = INHIBIT UNIT STATUS ON IOC
  Q-RSLT   44- 47     4 N       RESULT DESCRIPTOR STORAGE
  Q-DESC   48- 71    24 N       I/O DESCRIPTOR STORAGE
  Q-OP     48- 49     2 N       DESCRIPTOR OP CODE
  Q-OPV1       50     1 N       DESCRIPTOR VAR #1
  Q-OPV2       51     1 N       DESCRIPTOR VAR #2
  Q-OPV3   52- 53     2 N       DESCRIPTOR VAR #3
  Q-OP/A   54- 59     6 N       DESCRIPTOR BEGIN ADDRESS
  Q-OP/B   60- 65     6 N       DESCRIPTOR END   ADDRESS
  Q-OP/D   66- 71     6 N       DESCRIPTOR DISK  ADDRESS
 +Q-HLNK   72- 79     8 N       QUEUE ACCESS TABLE ENTRY ADDRESS (IOQ/IWOR)
 +Q-RAD    72- 79     8 N       ACTUAL ENDING ADDRESS VIA RAD (IOC)
 +Q-SVX1   72- 79     8 N       R/D ADDRESS (IX1) FOR SPURIOUS I/O
```

QUEUE BODY TABLE Q-====
(Continued)

Q-RDFG	80	1	N	RESULT DESCRIPTOR STATUS 0 = ERROR FREE IOC 1 = INVALID RESULT DESCRIPTOR 2 = INVALID I/O DESCRIPTOR 3 = UNIT PARITY ERROR 4 = MEMORY PARITY ERROR 5 = DEVICE NOT READY 6 = END OF FILE 7 = END OF UNIT MEDIA (IE: EOT) 8 = INTERFACE PARITY ERROR 9 = AVAILABLE A = SHORT RECORD IGNORED B = LONG RECORD IGNORED C = END OF PAGE IGNORED D = AVAILABLE E = SPECIFIED ERROR IGNORED F = ALL I/O ERRORS IGNORED
Q-TYP4	81	1	N	:8 MAINTENANCE LOG RECORD REQUIRED :4 SUBSEQUENT INITIATION OF I/O :2 ERROR RECOVERY OPERATION :1 I/O MUST BE INITIATED ON Q-CHAN
Q-CHAN	82- 83	2	N	INITIAL I/O CHANNEL
Q-LCPR	84- 99	16	N	16 DIGIT LCP RESULT DESCRIPTOR
Q-ERRD	100-115	16	N	16 DIGIT ERROR RESULT DESCRIPTOR
Q-ERCT	116-117	2	N	ERROR RETRY NUMBER
Q-HDW	118-119	2	N	HARDWARE TYPE FOR IOC ERROR IGNORE TEST
Q-RTRN	120-125	6	N	IOC EXIT RETURN ADDRESS (Q-EXIT=1)
Q-SEG#	126-127	2	N	MCP OVERLAY SEGMENT NUMBER TO RETURN TO
Q-VAR	128-129	2	N	VARIANT DIGITS FOR DISK/PACK IO DESCRIPTOR (MCPVI ONLY)
Q-ODLP	128-129	2	N	ORIGINAL ERROR DLP FOR MLOG (00.37) (MCPIX ONLY)
Q-PARM	130-143	14	N	SPECIAL PARAMETER STORAGE

DISK

Q-EU#	130-132	3	N	LOGICAL EU NUMBER FOR THIS DISK/PACK I/O
Q-LOCK	133	1	N	:8 FORCE ERROR FREE IOC :4 IGNORE PREVIOUS LOCK BIT :2 PERFORM READ CHECK (OP=52) ON DISK WRITE :1 QUEUE ELEMENT HAS RLT ENTRY ASSIGNED
Q-RWTF	134	1	N	DISK I/O TYPE CODE

READ/WRITE RQST:	MCP GENERATED RQST:
0 = READ	0 = READ/WRITE
1 = WRITE	1 = RESERVED
2 = SEEK	2 = READ LOCK
3 = RESERVED	3 = LOCK

QUEUE BODY TABLE Q-====
(Continued)

4 = LOCK	4 = RESERVED
5 = WRITE W/O UNLK	5 = WRITE W/O UNLK
6 = LOCK SEEK	6 = RESERVED
7 = RESERVED	7 = UNLK (SINGLE ADDRS)
8 = READ LOCK	8 = UNLK (ALL PROC ADDRS)
9 = UNLOCK	9 = RESERVED
A = SEEK LOCK	A = CLR (SINGLE ADDRS)
B = RESERVED	B = CLR (ALL PROC ADDRS)
C = READ RETRY ON LOCK	C = RESERVED
D = RESERVED	D = REPORT (SLOW)
E = READ RETRY ON LOCK (SEEK)	E = REPORT (FAST)
F = RESERVED	F = REPORT (QUICK)

Q-TIME	135-136	2	N	TIME ELEMENT FIRST ENTERED QUEUE (SHRD ONLY)
Q-OLDX	137	1	N	ORIGINAL VALUE OF Q-EXIT THAT WILL NEED TO BE PUT INTO SPECIAL UNLOCK-ONLY I/O
Q-DMS#	138-139	2	N	DMS USER INVOCATION NUMBER
Q-NEXT	140-143	4	N	LINK TO ORIGINAL Q ELEMENT IF THIS IS A SPECIAL LOCK-ONLY I/O (Q-TYP3:8)

DCOM

Q-DCTP	130-131	2	N	I/O TYPE CODE
Q-DCIO	132	1	N	TRANSLATE MODE (INPUT/OUTPUT)

MTPERR

Q-TERB	130-135	6	N	READ BACKWARD REMAINING BLOCK SIZE
Q-TERF	136-137	2	N	ERASE FORWARD UNIT SIZE (HUNDREDS)
Q-TERC	138	1	N	REMAINING ERASE BLOCK COUNT

SRTR

Q-MFIB	12- 19	8	N	SORTER FIB ADDRESS (ABSOLUTE)
Q-MRDA	66- 71	6	N	SORTER I/O RESULT DESCRIPTOR ADDRESS
Q-SOFT	72- 79	8	N	START OF SOFT INTERFACE AREA
Q-CWNT	128-129	2	N	TANKING SYNCRO COUNTER FOR 4A TYPE CONTROL
Q-TXLD	130-137	8	N	4A TYPE CONTROL TEXT LOAD END ADDRESS
Q-RSKF	138	1	N	:F = SORTER INTERFACE POINTERS NOT ABSOLUTE READ GIVEN FLAG
Q-TNKF	139	1	N	TANKING IN PROGRESS FLAG :8 JAM / MIS-SORT OR END-OF-FILE WHILE TANKING :4 RESERVED

QUEUE BODY TABLE Q-=====
(Continued)

				:2 RESERVED
				:1 TANKING FORCED
Q-FSVR	140	1	N	FLOW-STOP REQUESTED BY PROGRAM
Q-DMND	141	1	N	DEMAND MODE FLAG
				:8 RESERVED
				:4 RESERVED
				:2 DEMAND MODE POCKET SELECT IN PROGRESS
				:1 DEMAND MODE I/O IN PROGRESS

DCP

Q-DCEX	130-131	2	N	DCP MODULE INTERNAL IOC EXIT CODE
Q-DCBA	132-139	8	N	BASE OF THE BUFFER FROM WHICH THIS I/O WAS FIRED

DIRECT I/O

Q-RADB	136-143	8	N	BEGINNING "RAD" ADDRESS

QUEUE HEAD/TAIL TABLE QH-===

QH-NXT	0-		4	N	LINK TO NEXT QUEUE BODY ELEMENT
QH-LST	4-		4	N	LINK TO LAST QUEUE BODY ELEMENT
QH-LCH	8-		2	N	LINK (TIMES TEN) TO LOW CHANNEL ON EXCHANGE

QUEUE RESULT DESCRIPTOR TABLE QR-===

QR-R/D	0-		4	N	HARDWARE GENERATED RESULT DESCRIPTOR
QR-LNK	4-		4	N	SCAN RESULT DESCRIPTOR LINK
QR-XRD	8-	1	4	N	LCP R/D WORD (SECOND R/D WORD)
QR-EXC	12-	1	2	N	LINK (TIMES TEN) TO NEXT CHANNEL ON EXCHANGE
QR-BCT	14-	1	6	N	BRANCH COMMUNICATE ADDRESS
QR-INP	1560-	156	4	N	LINK TO IN-PROGRESS QUEUE-BODY ELEMENT
QR-IOC	1564-	156	6	N	I/O COMPLETE BRANCH ADDRESS
QR-LCP		157	1	N	
					:8 SUBSYSTEM POLL IN PROCESS
					:4 LCP CHANNEL
					:2 RESERVED
					:1 RESERVED
QR-SUB		157	1	N	DISK SUBSYSTEM TABLE INDEX
QR-TIM	1572-	157	8	N	TIME I/O INITIATED (TEMP)

RESULT DESCRIPTOR TABLE RD/===

This table contains those LCP Result Descriptor bits of interest and their nearest equivelant old-style R/D. As the LCP R/D may be from 8 to 16 digits on length and each case entry only contains 4 digits of R/D, in some cases it is necessary to use two case entries to cover all possible bits for a single old-style R/D. The table entries are organized as follows:

RD/OFF	0-1	2 N	FFSET TO FIRST R/D DIGIT OF INTEREST
RD/MSK	2-5	4 N	MASK BITS WHICH INDICATE R/D CASE IN LCP R/D
RD/FLG	6	1 N	FLAG BITS CONTROLLING INTERPRETATION OF R/D
			=@F@ BRANCH IF TEST OP (ADDR IN RD/OFF:6)
			:8 TERMINATE ANALYSIS IF CASE PRESENT
			:4 LAST CASE FOR HARDWARE/OP TYPE
			:2 MOVE BITS DETERMINED BY RD/MSK (TEST OP)
			:1/ BIT ONE TEST FOR MASK BITS PRESENT
			:1 BIT ZERO TEST FOR MASK BITS PRESENT (LIMITED TO FIRST TWO DIGITS OF MASK)
RD/OLD	7-9	3 N	OLD-STYLE R/D IF CASE PRESENT

RECORD LOCKOUT TABLE (SHRD) RLT===

The Record Lockout Table is used by the Shared Disk module to provide record level protection for files being updated by more than one program. When used with File Protect Memory hardware, it provides this protection for files being used by programs on different processors. The Record Lockout Table contains three basic types of entries:

1. Disk addresses which programs on our processor have currently locked,

2. disk addresses which programs on our processor are attempting to read, but which are currently locked by programs on another processor, and

3. disk addresses which programs on our processor are attempting to read, but which are currently locked of contended for by other programs on our processor.

RLT	0-19			RECORD LOCKOUT TABLE ENTRY
RLT-DA	0- 7	2	N	DISK ADDRESS LOCKED OR CONTENDED
RLT-MX	8- 9	2	N	MIX NUMBER OF LOCKING / CONTENDING PROGRAM
RLT-ST	10	1	N	LOCK I/O STATUS: :8 FPM OVERFLOW DETECTED DURING IOC :4 AVAILABLE :2 FPM CONTENTION :1 RLT CONTENTION
RLT-OP	11	1	N	TYPE OF USER I/O: 0 = WRITE 1 = READ 3 = READ WITH LOCK 4 = WRITE WITHOUT UNLOCK 5 = LOCK (READ LOCK NO DATA TRANSFER) 7 = READ ON PREVIOUSLY LOCKED ADDRESS 8 = UNLOCK
RLT-LK	12	1	N	LOCK STATUS: 0 = ADDRESS NOT LOCKED 1 = ADDRESS LOCKED
RLT-TM	13- 15	3	N	TIME (SEC) I/O QUEUED UP
RLT-QX	16- 19	4	N	Q-BODY INDEX OF I/O REQUEST (UNDIGIT "ADAD" INDICATES NO Q-BODY ATTACHED).

RUN LOG RECORD RL-===

Type	Digits	Description
0/0	100	File close record
0/1	100	File close remove
0/2	100	File removed
0/3	100	File title change
1/0	100	File open
2/0	100	End of job
3/0	200	Job schedule
4/0	100	Job schedule (MCP intrinsic)
4/1	100	Beginning of task (process call)
4/2	100	Usercode change
4/3	100	End of task (process return)
5/0	200	Log comments
5/1	200	Patch
5/2	200	Remote File Close
5/3	200	Remote File Open
6/0	100	Beginning of job
7/0	100	Idle time
7/1	100	Five minute accounting
7/2	100	Date change
7/3	100	Time change
7/4	100	Job RS'ed
7/5	100	Remote ODT log in/out
7/6	100	Job stopped
7/7	100	Job resumed
7/8	100	Job trailer
7/9	100	Job header
8/0	200	Halt/Load
8/1	200	Cold start, MCP loaded from tape
8/2	200	Warm start, MCP loaded from tape
8/3	200	Warm start, MCP loaded from disk
8/4	200	Warm start, MCP not reloaded
8/5	200	Halt load, MCP loaded from disk
8/6	200	Halt load, MCP automatic
8/7	200	Halt load, operator request
8/8	200	MCP log trailer
8/9	200	MCP log header
8/A	200	Cold start, MCP loaded from pack
8/B	200	Warm start, MCP loaded from pack
9/0	100	Filler record

RUN LOG RECORD

RL-===
(Continued)

Common to all Run Log Records

RL-TYP	00	01	N	LOG RECORD TYPE CODE
RL-***	01- 04	04	N	RESERVED
RL-ID#	05- 08	04	N	LOG ID NUMBER
RL-DTE	09- 14	06	N	DATE THIS RECORD CREATED MMDDYY
RL-TMS	15- 22	08	N	TIME STAMP FOR RECORD (MS)
	23	1	N	SUB-TYPE
	24- 99	38	A	LOG_REC_BODY {for 50-byte records or}
	24-199	88	A	LOG_REC_BODY {for 100-byte records}

```
***************************************************************************
* 100 Digits     FILE CLOSE RECORD          Type 0/0                       *
***************************************************************************
```

RL-SUB	23	01	N	SUB-TYPE = 0
RL-FID	24- 35	06	A	FILE ID
RL-MFD	36- 47	06	A	MULTI-FILE ID
RL-FL#	48- 49	02	N	FILE NUMBER (FIBFNM)
RL-CHN	50- 51	02	N	PRIMARY I/O CHANNEL
RL-UNT	52	01	N	UNIT NUMBER
RL-HDW	53- 54	02	N	HARDWARE TYPE CODE (FIBHDW)
RL-HDS	55	01	N	HARDWARE SUPPLEMENTARY TYPE CODE (IO-HDS)
RL-RL#	56- 58	03	N	TAPE REEL NUMBER (LABEL)
RL-PT#	59- 63	05	N	PHYSICAL TAPE NUMBER - MAG TAPE (LABEL)
RL-CTY	64- 65	02	N	CLOSE TYPE (COMMUNICATE PARAMETERS)
RL-RCT	66- 73	08	N	LOGICAL RECORD COUNT (FIBRCT)
RL-BCT	74- 81	08	N	PHYSICAL BLOCK COUNT (IO-BCT)
RL-ERR	82- 84	03	N	ERROR COUNT (FROM IOAT)
RL-NAR	85- 86	02	N	NUMBER OF DISK AREAS ACTUALLY USED
RL-EOF	87- 94	08	N	DISK END-OF-FILE POINTER (DF-EOF)
RL-DHC	95	01	N	DISK FILE HEADER BLOCK COUNT (CLOSE)
	96- 99	04	N	RESERVED

File close records are logged when a user program closes a file.

RUN LOG RECORD RL-===
(Continued)

```
*********************************************************************
* 100 Digits     FILE CLOSE REMOVE, FILE REMOVED    Types 0/1, 0/2  *
*********************************************************************
 RL-SUB       23  01 N    SUB-TYPE = 1 or 2
 RL-FID  24-  35  06 A    FILE ID
 RL-MID  36-  47  06 A    MULTI-FILE ID
         48-  49  02 N    RESERVED
 RL-CHN  50-  51  02 N    PRIMARY I/O CHANNEL
 RL-UNT       52  01 N    UNIT NUMBER
 RL-HDW  53-  54  02 N    HARDWARE TYPE (DISK=06/PACK=11)
 RL-HDS       55  01 N    HARDWARE SUB-TYPE CODE (IO-HDS)
         56-  79  12 A    RESERVED
 RL-UKY  80-  85  06 N    KEY TO RESPONSIBLE USER
         86-  99  14 N    RESERVED
```

The file close remove record indicates that a previous disk or pack file was removed by this close. The file removed record indicates the file was removed by the REMOVE control instruction or <mix>RM keyboard command.

```
*********************************************************************
* 100 Digits     FILE TITLE CHANGE          Type 0/3                *
*********************************************************************
 RL-SUB       23  01 N    SUB-TYPE = 3
 RL-FID  24-  35  06 A    OLD FILE ID
 RL-MID  36-  47  06 A    OLD MULTI-FILE ID
         48-  49  02 N    RESERVED
 RL-CHN  50-  51  02 N    PRIMARY I/O CHANNEL
 RL-UNT       52  01 N    UNIT NUMBER
 RL-HDW  53-  54  02 N    HARDWARE TYPE (DISK=06/PACK=11)
 RL-HDS       55  01 N    HARDWARE SUB-TYPE CODE   (IO-HDS)
 RL-FD2  56-  67  06 A    NEW FILE ID
 RL-MF2  68-  79  06 A    NEW MULTI-FILE ID
 RL-UKY  80-  85  06 N    KEY TO RESPONSIBLE USER
         86-  99  14 N    RESERVED
```

The file title change record is used to log the fact that a file title has been changed on disk or pack.

RUN LOG RECORD RL-===
(Continued)

```
**********************************************************************
* 100 Digits     FILE OPEN RECORD          Type 1/0                 *
**********************************************************************
 RL-SUB        23   01 N    SUB-TYPE = 0
 RL-FID   24-  35   06 A    FILE ID
 RL-MFD   36-  47   06 A    MULTI-FILE ID
 RL-FL#   48-  49   02 N    FILE NUMBER (FIBFNM)
 RL-CHN   50-  51   02 N    PRIMARY I/O CHANNEL
 RL-UNT        52   01 N    UNIT NUMBER
 RL-HDW   53-  54   02 N    HARDWARE TYPE CODE (FIBHDW)
 RL-HDS        55   01 N    HARDWARE SUPPLEMENTARY TYPE CODE (IO-HDS)
 RL-RL#   56-  58   03 N    TAPE REEL NUMBER                  (LABEL)
 RL-CRE   59-  63   05 N    FILE CREATION DATE     YYDDD      (LABEL)
 RL-CYC   64-  65   02 N    CYCLE NUMBER                      (LABEL)
 RL-MRL   66-  70   05 N    MAXIMUM RECORD LENGTH (FIBMRL)
 RL-RPB   71-  73   03 N    RECORDS PER BLOCK (FIBRPB)
 RL-MBS   74-  79   06 N    MAXIMUM BLOCK SIZE (FIBMBS)
 RL-BLA        80   01 N    BUFFER ACCESS TECHNIQUE (FIB-BA)
 RL-LBL        81   01 N    FILE LABEL CONVENTION (FIBLBL)
 RL-ALT        82   01 N    NUMBER OF ALTERNATE AREAS (FIBALT)
 RL-IOF        83   01 N    INPUT/OUTPUT FLAG (FIB-IO)
 RL-MOD        84   01 N    EXTERNAL RECORDING MODE (FIBMOD)
 RL-BLK        85   01 N    BLOCKING TECHNIQUE (FIBBLK)
 RL-SPF        86   01 N    SPECIAL FORMS FLAG (FIBSPF)
 RL-SVF   87-  89   03 N    SAVE FACTOR IN DAYS (FIB-SV)
 RL-DKA   90-  96   07 N    DISK SEGMENTS PER AREA ( DF-DSA)
 RL-DSA        97   01 N    DISK ACCESS TECHNIQUE (FIBDTK)
 RL-DHO        98   01 N    DISK FILE HEADER BLOCK COUNT  (OPEN)
 RL-USE        99   01 N    FILE USE TYPE (IO-USE)
```

A file open record is logged when a program opens a file of any type.

RUN LOG RECORD RL-===
(Continued)

```
******************************************************************
* 100 Digits    END OF JOB RECORD         Type 2/0               *
******************************************************************
 RL-SUB       23  01 N    SUB-TYPE = 0
 RL-PID  24-  35  06 A    PROGRAM ID
 RL-MID  36-  47  06 A    PROGRAM MF-ID
 RL-JOB  48-  49  02 N    JOB NUMBER
 RL-CHN  50-  51  02 N    PRIMARY I/O CHANNEL -- REMOTE SPO USER
 RL-UNT       52  01 N    UNIT NUMBER
 RL-HDW  53-  54  02 N    HARDWARE TYPE CODE (FIBHDW-ZERO IF NONE)
 RL-HDS       55  01 N    HARDWARE SUPPLEMENTARY TYPE CODE (IO-HDS)
 RL-OVL  56-  61  06 N    PROGRAM OVERLAY COUNT
 RL-TRM  62-  63  02 N    FINISH CODE (MIX-TC)
 RL-KOR  64-  66  03 N    MEMORY RQD (KD) - W/O DFHDR
         67-  73  07 N    RESERVED
 RL-PDT  74-  81  08 N    PROGRAM DIRECT PROCESSOR TIME      (MS)
 RL-PPT  82-  89  08 N    PROGRAM PRORATED PROCESSOR TIME    (MS)
 RL-PWT  90-  97  08 N    PROGRAM WTG I/O ACCUM TIME         (MS)
 RL-DSC  98-  99  02 N    USER DS CODE  (DEFAULT = ZERO)
```

An end of job record is logged when a job terminates.

RUN LOG RECORD RL-===
(Continued)

```
************************************************************************
* 200 Digits     JOB SCHEDULE RECORD        Types 3/0                  *
************************************************************************
  RL-SUB        23  01 N    SUB-TYPE = 0
  RL-PID   24-  35  06 A    PROGRAM ID
  RL-MID   36-  47  06 A    PROGRAM MF-ID
  RL-JOB   48-  49  02 N    JOB NUMBER
  RL-DSG   50-  55  06 N    DISK SEGMENTS IN PROGRAM
  RL-CG#   56-  61  06 N    USER CHARGE NUMBER
  RL-COR   62-  64  03 N    MEMORY RQD (KD) - W/O DISK FILE HEADERS
  RL-FLS   65-  66  02 N    NUMBER OF FILES
           67-  68  02 N    RESERVED
  RL-EXC        69  01 N    JOB EXECUTION CODE (MIX-PI)
  RL-SCH        70  01 N    SCHEDULE TYPE CODE
                              0 IN THE MIX
                              1 JOB IN SCHEDULE
                              2 EXECUTE AFTER
  RL-SXE        71  01 N    SUPPLEMENTARY EXECUTION CODE (MIX-FG)
  RL-INT        72  01 N    MCP INTRINSIC FLAG
                73  01 N    RESERVED
  RL-KEY   74-  79  06 N    KEY TO RESPONSIBLE USER
  RL-HDR   80-199   60 A    HEADER INFORMATION FROM CONTROL CARD
```

A job schedule record is logged when a program is entered into the MCP job schedule for execution. The field RL-HDR contains the control text that invoked scheduling of the job. The type 3:0 long schedule records are used to log the scheduling of all jobs except MCP intrinsics.

RUN LOG RECORD

RL-===
(Continued)

```
***********************************************************************
* 100 Digits    JOB SCHEDULE RECORD        Types 4/0                  *
***********************************************************************
```

RL-SUB	23	01	N	SUB-TYPE = 0
RL-PID	24- 35	06	A	PROGRAM ID
RL-MID	36- 47	06	A	PROGRAM MF-ID
RL-JOB	48- 49	02	N	JOB NUMBER
RL-DSG	50- 55	06	N	DISK SEGMENTS IN PROGRAM
RL-CG#	56- 61	06	N	USER CHARGE NUMBER
RL-COR	62- 64	03	N	MEMORY RQD (KD) - W/O DISK FILE HEADERS
RL-FLS	65- 66	02	N	NUMBER OF FILES
	67- 68	02	N	RESERVED
RL-EXC	69	01	N	JOB EXECUTION CODE (MIX-PI)
RL-SCH	70	01	N	SCHEDULE TYPE CODE (SEE TYPE 3/0)
RL-SXE	71	01	N	SUPPLEMENTARY EXECUTION CODE (MIX-FG)
RL-INT	72	01	N	MCP INTRINSIC FLAG
	73	01	N	RESERVED
RL-KEY	74- 79	06	N	KEY TO RESPONSIBLE USER
	80- 99	20	N	RESERVED

A job schedule record is logged when a program is entered into the MCP job schedule for execution. Type 4/0, short schedule records, are used to log the scheduling of MCP intrinsics.

```
***********************************************************************
* 100 Digits    BEGINNING OF TASK (PROCESS CALL)  Type 4/1            *
***********************************************************************
```

RL-SUB	23	01	N	SUB-TYPE = 1
RL-PID	24- 35	06	A	PROGRAM ID
RL-MID	36- 47	06	A	PROGRAM MF-ID
RL-JOB	48- 49	02	N	JOB NUMBER
RL-DSG	50- 55	06	N	RESERVED
RL-CG#	56- 61	06	N	USER CHARGE NUMBER
RL-COR	62- 64	03	N	MEM REQD (KD) - W/O DISK FILE HEADERS
	65- 73	09	N	RESERVED
RL-KEY	74- 79	06	N	KEY TO RESPONSIBLE USER
	80- 99	20	N	RESERVED

A beginning of task record is logged when a program calls a task. A task is any program that is called by the process call MCP Communicate. An example for CANDE would be a call on RCSPBD from EDITOR.

These records are not logged if the NOCALL option is specified for the MCP option, USE RLOG.

RUN LOG RECORD RL-===
(Continued)

```
**********************************************************************
* 100 Digits    USERCODE CHANGE             Type 4/2                 *
**********************************************************************
 RL-SUB       23   01 N     SUB-TYPE = 2
 RL-PID  24-  35   06 A     PROGRAM ID
 RL-MID  36-  47   06 A     PROGRAM MF-ID
 RL-JOB  48-  49   02 N     JOB NUMBER
 RL-DSG  50-  55   06 N     RESERVED
 RL-CG#  56-  61   06 N     USER CHARGE NUMBER
 RL-COR  62-  64   03 N     MEM REQD (KD) - W/O DISK FILE HEADERS
         65-  73   09 N     RESERVED
 RL-KEY  74-  79   06 N     KEY TO RESPONSIBLE USER
         80-  99   20 N     RESERVED
```

A usercode change record is logged when a program requests and receives a change in the usercode under which it is running from the MCP.

```
**********************************************************************
* 100 Digits    END OF TASK (PROCESS RETURN)    Type 4/3             *
**********************************************************************
 RL-SUB       23   01 N     SUB-TYPE = 3
 RL-PID  24-  35   06 A     PROGRAM ID
 RL-MID  36-  47   06 A     PROGRAM MF-ID
 RL-JOB  48-  49   02 N     JOB NUMBER
 RL-DSG  50-  55   06 N     RESERVED
 RL-CG#  56-  61   06 N     USER CHARGE NUMBER
 RL-COR  62-  64   03 N     MEM REQD (KD) - W/O DISK FILE HEADERS
         65-  73   09 N     RESERVED
 RL-KEY  74-  79   06 N     KEY TO RESPONSIBLE USER
         80-  99   20 N     RESERVED
```

An end of task record is logged when a task terminates and returns to the program that called it. A task is any program that is initiated by the process call MCP Communicate. An example for CANDE would be the return of RCSPBD to EDITOR.

These records are not logged if NOCALL was specified for the MCP option, USE RLOG.

RUN LOG RECORD RL-===
(Continued)

```
*********************************************************************
* 200 Digits     LOG COMMENT RECORD       Type 5/0                  *
*********************************************************************
  RL-SUB        23  01 N    SUB-TYPE = 0   LOG ID MAY BE ZERO
  RL-PID   24- 35  06 A    PROGRAM ID
  RL-MID   36- 47  06 A    PROGRAM MF-ID
  RL-JOB   48- 49  02 N    JOB NUMBER
  RL-CMT   50-199  75 A    OPERATOR LOG COMMENT
```

The Log comment record is used to log operator comments entered by the LCR ODT command. The text is logged exactly as entered.

```
*********************************************************************
* 200 Digits     PATCH RECORD             Type 5/1                  *
*********************************************************************
  RL-SUB        23 01 N    SUB-TYPE = 1
  RL-PID   24- 35 06 A    FILE-ID
  RL-MID   36- 47 06 A    PACK-ID (PATCH TO DISKPACK)
  RL-PHW   48- 49 02 N    HARDWARE TYPE DISK/PACK (06/11)
  RL-PSG   50- 61 12 N    PATCH SEGMENT (XNN OR DISK ADDR)
  RL-SAD   62- 67 06 N    SEGMENT STARTING ADDRESS
  RL-PAD   68- 73 06 N    PATCH STARTING ADDRESS
  RL-PTY        74 01 N    PATCH TYPE (0 = UN, 2 = UA)
  RL-PSZ   75- 77 03 N    PATCH SIZE IN PATCH UNITS (UN/UA)
  RL-KEY   78- 83 06 N    KEY TO RESPONSIBLE USER (NOT SET)
  RL-PTX   84-143 30 A    PATCH TEXT IN ALPHA
          144-199 56 N    RESERVED
```

The patch record is for logging all patches to files and programs in the system. One record occurs for each patch entered.

RUN LOG RECORD RL-===
(Continued)

```
***********************************************************************
* 200 Digits      REMOTE FILE CLOSE RECORD       Type 5/2             *
***********************************************************************
 RL-SUB         23 01 N     SUB-TYPE = 2
 RL-FID     24- 35 06 A     FILE ID
 RL-MFD     36- 47 06 A     MULTI-FILE ID
 RL-FL#     48- 49 02 N     FILE NUMBER (FIBFNM)
            50- 52 03 N     RESERVED
 RL-HDW     53- 54 02 N     HARDWARE TYPE CODE (FIBHDW)
 RL-HDS         55 01 N     HARDWARE SUPPLEMENTARY TYPE CODE (IO-HDS)
 RL-RL#     56- 58 03 N     TAPE REEL NUMBER                 (LABEL)
 RL-PT#     59- 63 05 N     PHYSICAL TAPE NUMBER - MAG TAPE (LABEL)
 RL-CTY     64- 65 02 N     CLOSE TYPE (COMMUNICATE PARAMETERS)
 RL-RCT     66- 73 08 N     LOGICAL RECORD COUNT (FIBRCT)
 RL-BCT     74- 81 08 N     PHYSICAL BLOCK COUNT (IO-BCT)
 RL-ERR     82- 84 03 N     ERROR COUNT (FROM IOAT)
 RL-NAR     85- 86 02 N     NUMBER OF DISK AREAS ACTUALLY USED
            87- 94 08 N     RESERVED
 RL-DHC         95 01 N     DISK FILE HEADER BLOCK COUNT (CLOSE)
            96- 99 04 N     RESERVED
 RL-HSN    100-133 17 A     REMOTE HOSTNAME
           134-199 66 N     RESERVED
```

```
        RUN LOG RECORD                                          RL-===
                                                            (Continued)

     ***************************************************************************
     * 200 Digits     REMOTE FILE OPEN RECORD     Type 5/3                     *
     ***************************************************************************
      RL-SUB        23 01 N    SUB-TYPE = 3
      RL-FID   24-  35 06 A    FILE-ID
      RL-MFD   36-  47 06 A    MULTI-FILE ID
      RL-FL#   48-  49 02 N    FILE NUMBER (FIBFNM)
               50-  57 03 N    RESERVED
      RL-HDW   53-  54 02 N    HARDWARE TYPE CODE (FIBHDW)
      RL-HDS        55 01 N    HARDWARE SUPPLEMENTARY TYPE CODE (IO-HDS)
      RL-RL#   56-  58 03 N    TAPE REEL NUMBER                 (LABEL)
      RL-CRE   59-  63 05 N    FILE CREATION DATE               (LABEL)
      RL-CYC   64-  65 02 N    CYCLE NUMBER                     (LABEL)
      RL-MRL   66-  70 05 N    MAXIMUM RECORD LENGTH (FIBMRL)
      RL-RPB   71-  73 03 N    RECORDS PER BLOCK (FIBRPB)
      RL-MBS   74-  79 06 N    MAXIMUM BLOCK SIZE (FIBMBS)
      RL-BFA        80 01 N    BUFFER ACCESS TECHNIQUE (FIB-BA)
      RL-LBL        81 01 N    FILE LABEL CONVENTION (FIBLBL)
      RL-ALT        82 01 N    NUMBER OF ALTERNATE AREAS (FIBALT)
      RL-IOF        83 01 N    INPUT/OUTPUT FLAG (FIB-IO)
      RL-MOD        84 01 N    EXTERNAL RECORDING MODE (FIBMOD)
      RL-BLK        85 01 N    BLOCKING TECHNIQUE (FIBBLK)
      RL-SPF        86 01 N    SPECIAL FORMS FLAG (FIBSPF)
      RL-SVF   87-  89 03 N    SAVE FACTOR IN DAYS (FIB-SV)
               90-  96 07 N    RESERVED
      RL-DKA        97 01 N    DISK ACCESS TECHNIQUE (FIBDTK)
                    98 01 N    RESERVED
      RL-USE        99 01 N    FILE USE TYPE   (IO-USE)
      RL-HSN  100-133 17 A    REMOTE HOSTNAME
              134-199 66 N    RESERVED
```

RUN LOG RECORD   RL-===
(Continued)

```
***********************************************************************
* 100 Digits      BEGINNING OF JOB RECORD    Type 6/0                 *
***********************************************************************
  RL-SUB          23  01 N    SUB-TYPE = 0
  RL-PID     24-  35  06 A    PROGRAM ID
  RL-MID     36-  47  06 A    PROGRAM MF-ID
  RL-JOB     48-  49  02 N    JOB NUMBER
  RL-DSG     50-  55  06 N    DISK SEGMENTS IN PROGRAM
  RL-CG#     56-  61  06 N    USER CHARGE NUMBER
  RL-COR     62-  64  03 N    MEMORY RQD (KD) - W/O DISK FILE HEADERS
  RL-FLS     65-  66  02 N    NUMBER OF FILES
             67-  68  02 N    RESERVED
  RL-EXC          69  01 N    JOB EXECUTION CODE   (MIX-PI)
                  70  01 N    RESERVED
  RL-SXE          71  01 N    SUPPLEMENTARY EXECUTION CODE  (MIX-FG)
  RL-INT          72  01 N    MCP INTRINSIC FLAG
                  73  01 N    RESERVED
  RL-DTC     74-  79  06 N    DATE PROGRAM COMPILED (BOJ ONLY) MMDDYY
             80-  99  20 N    RESERVED
```

The beginning of job record is logged when a schedule job starts execution. Exceptions to this are swap area tasks.

```
***********************************************************************
*           TYPE  7             100 DIGITS   MISC. ACTG.              *
***********************************************************************
  RL-SUB          23  01 N    SUB-TYPE CODE
                              0 - IDLE-TIME (5 MINUTES OR LESS)
                              1 - FIVE-MINUTE PRORATED-TIME TOTALS
                              2 - OPERATOR CHANGED DATE
                              3 - OPERATOR CHANGED TIME
                              4 - JOB REMOVED FROM SCHEDULE
                              5 - REMOTE SPO LOG-IN/LOG-OUT
                              6 - JOB EXECUTION STOPPED
                              7 - JOB EXECUTION RESUMED
                              8 - JOB LOG TRAILER
                              9 - JOB LOG HEADER
```

RUN LOG RECORD RL-===
(Continued)

```
*************************************************************************
* 100 Digits      IDLE TIME RECORD          Type 7/0                     *
*************************************************************************
  RL-SUB         23   01 N     SUB-TYPE = 0      LOG ID IS ZERO
  RL-IDL   24-   31   08 N     IDLE-TIME (MS)
           32-   61   30 N     RESERVED
  RL-COR   62-   64   03 N     MCP MEMORY  IN USE   KB
           65-   99   35 N     RESERVED
```

Processor idle time records log the amount of processor idle time that has accumulated since the last idle time record.

```
*************************************************************************
* 100 Digits      FIVE MINUTE ACCOUNTING RECORD Type 7/1                 *
*************************************************************************
  RL-SUB         23   01 N     SUB-TYPE = 1     LOG ID IS ZERO
  RL-XDT   24-   29   06 N     TOTAL PROCESS TIME LAST FIVE MINUTES   MS
  RL-XPT   30-   35   06 N     TOTAL PRORATE TIME LAST FIVE MINUTES   MS
  RL-5WT   36-   43   08 N     TOTAL WTG I/O TIME LAST FIVE MINUTES   MS
  RL-5IT   44-   49   06 N     TOTAL MCP IDLE TIME                    MS
           50-   61   12 N     RESERVED
  RL-COR   62-   64   03 N     MCP MEMORY IN USE                      KB
           65-   99   35 N     RESERVED
```

The five minute accounting record is used to log the processor usage for the last five minutes and will occur at five-minute intervals in the Run log.

```
*************************************************************************
* 100 Digits      DATE CHANGE RECORD        Type 7/2                     *
*************************************************************************
  RL-SUB         23   01 N     SUB-TYPE = 2       LOG ID IS ZERO
  RL-NDT   24-   29   06 N     NEW DATE (MMDDYY)
           30-   73   44 N     RESERVED
  RL-KEY   74-   79   06 N     KEY TO RESPONSIBLE USER
           80-   99   20 N     RESERVED
```

Date change records are logged when the operator changes the system date by the DR ODT command.

RUN LOG RECORD RL-===
(Continued)

```
*********************************************************************
* 100 Digits     TIME CHANGE RECORD        Type 7/3                 *
*********************************************************************
 RL-SUB       23  01 N    SUB-TYPE = 3          LOG ID IS ZERO
 RL-NDT  24-  29  06 N    NEW TIME (SECONDS FROM MIDNIGHT)
         30-  73  44 N    RESERVED
 RL-KEY  74-  79  06 N    KEY TO RESPONSIBLE USER
         80-  99  20 N    RESERVED
```

The time change record is logged when the operator changes the system time by the TR ODT command.

```
*********************************************************************
* 100 Digits     RS-ED JOB RECORD          Type 7/4                 *
*********************************************************************
 RL-SUB       23  01 N    SUB-TYPE = 4
 RL-PID  24-  35  06 A    PROGRAM ID
 RL-MID  36-  47  06 A    PROGRAM MF-ID
 RL-JOB  48-  49  02 N    JOB NUMBER
         50-  73  24 N    RESERVED
 RL-KEY  74-  79  06 N    KEY TO RESPONSIBLE USER
         80-  99  20 N    RESERVED
```

The job RSed record is logged when a program has been removed from the MCP job schedule abnormally by the RS ODT command.

```
*********************************************************************
* 100 Digits     REMOTE ODT LOG IN/OUT RECORD  Type 7/5             *
*********************************************************************
 RL-SUB       23  01 N    SUB-TYPE = 5     LOG ID IS ZERO
 RL-FID  24-  35  06 A    RESERVED
 RL-MFD  36-  47  06 A    ADAPTER ID
              48  01 N    RESERVED
 RL-FL#       49  01 N    LOG IN/OUT INDICATOR (0=IN, 1=OUT)

         THE FOLLOWING FIELDS APPLY TO THE LOG-IN RECORD ONLY
 RL-CHN  50-  51  02 N    REMOTE SPO I/O CHANNEL #
 RL-UNT       52  01 N    REMOTE SPO UNIT #
 RL-HDW  53-  54  02 N    REMOTE SPO HARDWARE TYPE CODE (FIBHDW)
 RL-HDS       55  01 N    SUPPLEMENTARY HARDWARE CODE   (IO-HDS)
 RL-CG#  56-  61  06 N    REMOTE USER CHARGE #
 RL-KEY  62-  67  06 N    SECURITY ID
         68-  99  32 N    RESERVED
```

The remote ODT record is logged when a remote ODT is logged in or out by the ?LI or ?LO OCL commands.

RUN LOG RECORD RL-===
(Continued)

```
**********************************************************************
* 100 Digits     JOB STOPPED RECORD       Type 7/6                   *
**********************************************************************
 RL-SUB        23  1 N    SUB-TYPE = 6
 RL-PID    24- 35  6 A    PROGRAM ID
 RL-MID    36- 47  6 A    PROGRAM MF-ID
 RL-JOB    48- 49  2 N    JOB NUMBER
 RL-DSG    50- 55  6 N    DISK SEGMENTS USED FOR ROLL-OUT
 RL-CG#    56- 61  6 N    RESERVED
 RL-COR    62- 64  3 N    MEMORY RETURNED  KD
               65  1 N    RESERVED
 RL-DHH    66- 68  3 N    DISK FILE HEADER BLOCK COUNT  (TOTAL)
           69- 73  5 N    RESERVED
 RL-KEY    74- 79  6 N    KEY TO RESPONSIBLE USER
           80- 99 20 N    RESERVED
```

The job stopped record is logged when ever a job is stopped.

These records are not logged if NOSTGO was specified for the MCP option, USE RLOG.

```
**********************************************************************
* 100 Digits     JOB RESUMED RECORD       Type 7/7                   *
**********************************************************************
 RL-SUB        23  1 N    SUB-TYPE = 7
 RL-PID    24- 35  6 A    PROGRAM ID
 RL-MID    36- 47  6 A    PROGRAM MF-ID
 RL-JOB    48- 49  2 N    JOB NUMBER
 RL-DSG    50- 55  6 N    DISK SEGMENTS USED FOR ROLL-OUT
 RL-CG#    56- 61  6 N    RESERVED
 RL-COR    62- 64  3 N    MEMORY RQD (KD) - W/O DISK FILE HEADERS
               65  1 N    RESERVED
 RL-DHH    66- 68  3 N    DISK FILE HEADER BLOCK COUNT  (TOTAL)
           69- 73  5 N    RESERVED
 RL-KEY    74- 79  6 N    KEY TO RESPONSIBLE USER
           80- 99 20 N    RESERVED
```

A job resumed record is logged when a job is restored to memory after being stopped and is allowed to begin running again.

These records are not logged if NOSTGO was specified for the MCP option, USE RLOG.

RUN LOG RECORD RL-===
(Continued)

```
*******************************************************************
* 100 Digits      JOB TRAILER RECORD        Type 7/8              *
*******************************************************************
```

RL-SUB	23	1	N	SUB-TYPE = 8
RL-PID	24- 35	6	A	PROGRAM ID
RL-MID	36- 47	6	A	PROGRAM MF-ID
RL-JOB	48- 49	2	N	JOB NUMBER
RL-DSG	50- 55	6	N	RESERVED
RL-CG#	56- 61	6	N	USER CHARGE NUMBER
RL-COR	62- 64	3	N	MEMORY RQD (KD) - W/O DISK FILE HEADERS
RL-FLS	65- 66	2	N	NUMBER OF FILES
	67- 68	2	N	RESERVED
RL-EXC	69	1	N	JOB EXECUTION CODE (MIX-PI)
RL-SCH	70	1	N	SCHEDULE TYPE CODE (SEE TYPE 3/0)
RL-SXE	71	1	N	SUPPLEMENTARY EXECUTION CODE (MIX-FG)
RL-INT	72	1	N	MCP INTRINSIC FLAG
	73	1	N	RESERVED
RL-PDT	74- 81	8	N	PROGRAM DIRECT PROCESSOR TIME (MS)
RL-PPT	82- 89	8	N	PROGRAM PRORATED PROCESSOR TIME (MS)
RL-PWT	90- 97	8	N	PROGRAM WTG I/O ACCUM TIME (MS)
	98- 99	2	N	RESERVED

This record is a pseudo end of job record, and one is written for each job in the mix during log transfer.

```
*******************************************************************
* 100 Digits      JOB HEADER RECORD         Type 7/9              *
*******************************************************************
```

RL-SUB	23	1	N	SUB-TYPE = 9
RL-PID	24- 35	6	A	PROGRAM ID
RL-MID	36- 47	6	A	PROGRAM MF-ID
RL-JOB	48- 49	2	N	JOB NUMBER
RL-DSG	50- 55	6	N	DISK SEGMENTS IN PROGRAM
RL-CG#	56- 61	6	N	USER CHARGE NUMBER
RL-COR	62- 64	3	N	MEMORY RQD (KD) - W/O DISK FILE HEADERS
RL-FLS	65- 66	2	N	NUMBER OF FILES
	67- 68	2	N	RESERVED
RL-EXC	69	1	N	JOB EXECUTION CODE (MIX-PI)
RL-SCH	70	1	N	SCHEDULE TYPE CODE (SEE TYPE 3/0)
RL-SXE	71	1	N	SUPPLEMENTARY EXECUTION CODE (MIX-FG)
RL-INT	72	1	N	MCP INTRINSIC FLAG
	73	1	N	RESERVED
RL-PDT	74- 81	8	N	PROGRAM DIRECT PROCESSOR TIME (MS)
RL-PPT	82- 89	8	N	PROGRAM PRORATED PROCESSOR TIME (MS)
RL-PWT	90- 97	8	N	PROGRAM WTG I/O ACCUM TIME (MS)
	98- 99	2	N	RESERVED

RUN LOG RECORD RL-===

(Continued)

The job header record is logged after a log transfer for each job active at the time of the transfer, and is a pseudo beginning of job record.

```
        RUN LOG RECORD                                    RL-===
                                                           (Continued)

     ***********************************************************************
     * 100 Digits     HALT LOAD RECORD          Types 8/0 - 8/3             *
     ***********************************************************************
      RL-SUB         23    1 N    SUB-TYPE CODE
                                  0 - HALT/LOAD
                                  1 - COLDSTART : MCP TAPE LOAD
                                  2 - WARMSTART : MCP TAPE LOAD
                                  3 - WARMSTART : MCP DISK LOAD
                                  4 - WARMSTART : NO MCP LOAD
                                  5 - HALT/LOAD : MCP DISK LOAD
                                  6 - HALT/LOAD : MCP ABORTED
                                  7 - HALT/LOAD : OPERATOR REQUEST
                                  8 - MCP LOG TRAILER
                                  9 - MCP LOG HEADER
                                  A - COLDSTART : MCP DISKPACK LOAD
                                  B - WARMSTART : MCP DISKPACK LOAD
      RL-PID     24- 35    6 A    MCP ID
      RL-MID     36- 47    6 A    RESERVED
      RL-ASR     48- 55    4 A    RELEASE ASR # (XX.X)
      RL-CG#     56- 61    6 N    DEFAULT CHARGE NUMBER
      RL-COR     62- 64    3 N    MCP MEMORY REQUIREMENTS
      RL-MEM     65- 68    4 N    TOTAL SYSTEM MEMORY
      RL-PC#         69    1 N    SYSTEM PROCESSOR NUMBER
      RL-CPU         70    1 N    PROCESSOR TYPE CODE
                                   2  MS-0
                                   3  MS-0
                                   4  MS-1  B4700
                                   5  MS-2  B4800
                                   6  B2900
      RL-MSP         71    1 N    SYSTEM MEMORY SPEED CODE
                                   0  1 MHZ
                                   1  2 MHZ
                                   2  3 MHZ
                                   3  4 MHZ
                                   4  6 MHZ
                                   5  8 MHZ
                                   6  7 MHZ
                 72- 73    2 N    RESERVED
      RL-DTC     74- 79    6 N    MCP VERSION DATE   (mmddyy)
      RL-MIT     80- 89   10 N    MCP IDLE TIME (LOG TRAILER)  (msec)
                 90- 99   10 N    RESERVED
```

The halt/load record is logged each time one of the following occurs, halt/load, coldstart, warmstart, or log transfer. The sub type specifies the type of function performed. The header record is the first record in the new log after a log transfer, and a trailer record is the last record in the old log after log transfer. If the log

RUN LOG RECORD RL-===
(Continued)

was initialized by a coldstart, then the first record in the log will be a coldstart type record. Normally these record types will occur only at the beginning or end of the log file, unless a halt/load or warmstart occurs in between. The EOF on the log file is determined by the occurrence of a type 8/8 log trailer record. Any subsequent records in the log file are invalid.

```
**********************************************************************
* 100 Digits     FILLER RECORD                Types 9/0              *
**********************************************************************
        24-99    38 A    RESERVED
```

The filler record is used to fill out the second half of a sector (50 bytes) so that the next record will start on a disk (100-byte) sector boundary. This record is used when the next log record to be written is a long record, 100 bytes, as it must start at the begining of a sector (100-byte). Only two 50-byte records or one 100-byte record will be put in a single sector.

```
          SEGMENT DICTIONARY                                     S-====

S-DICT     0- 13          SEGMENT DICTIONARY ENTRY FOR MCP OVERLAYS
S-SEG      0-  1   2 N    MCP OVERLAY SEGMENT NUMEER
S-CNT      2-  3   2 N    OVERLAY CALL COUNT
S-ADD      4-  7   4 N    QUICK OVERLAY MEMORY ADDRESS IN KD
S-STAT         8   1 N    QUICK OVERLAY STATUS:  0 = LOAD OVLY FROM MEMORY
                            :8 INHIBIT QUICK OVERLAY
                            :4 NO PRIORITY FOR QUICK OVERLAY
                            :2 READ OVERLAY FROM DISK - DO NOT STORE
                            :1 READ OVERLAY FROM DISK - STORE IN QWK OVLY
S-DISK     9- 13   5 N    LOW ORDER DISK ADDRESS OF MCP OVERLAY
```

SECURITY ATTRIBUTES STORAGE AREA SA-===

The Security Attributes Storage Area (SASA) is a 25-byte field which is generated by the compiler whenever security attributes are declared for a file. A file's security attributes are stored in the SASA. These attributes may come from label equate information, program specified information, or the disk directory. The source of the information depends upon the type of open performed on the file and on file equate parameters supplied with the execution.

The SASA is addressed using the pointer stored in FIBEXT. If no SASA is present, FIBEXT must be zero.

The SASA has the following layout.

SA-REV	0- 1	1	A	Revision level of SASA (UA)
SA-GRD	2- 13	6	A	Guard file ID
SA-FAM	14- 25	6	A	Pack name for guard file (default = DISK)
SA-STY	26	1	N	Security type 4 = Guarded 2 = Public 1 = Private (default) F = None
SA-SUS	27	1	N	Security use 6 = IO (default) 4 = IN 2 = OUT 1 = SECURED
SA-SNS	28	1	N	Sensitive data flag 0 = Not sensitive (default) 1 = Overwrite data with random pattern
SA-MAI	29	1	N	Reserved
SA-UCO	30- 49	10	A	Usercode of the creator

```
          DCP DISK MCPNIF FILE FORMAT                         SF-===

SF-MCS     0- 11   6 A    MCS NAME IF SF-DUM = (0 OR @F@)
                          FIRMWARE NAME IF SF-DUM = @E@
SF-DUM        12   1 N    RECORD TYPE
                          0 =  STATION RECORD
                          E =  DCP RECORD
                          F =  MCS FILE RECORD
SF-ND#        13   1 N    DCP ID NUMBER (STATION AND DCP RECORDS ONLY)
              14   1 N    NDL FLAGS (NOT USED BY MCP)
                            :8   MYUSE INPUT BOOLEAN FROM NDL
                            :4   MYUSE OUTPUT BOOLEAN FROM NDL
                            :2   ENABLE INPUT BOOLEAN FROM NDL
                            :1   RESERVED
              15   1 N    RESERVED
SF-PSN    16- 19   4 N    LINE/STATION ADDRESS (PHYSICAL STATION NUMBER)
                          FOR STATION RECORD
                          LARGEST S-MEMORY ADDRESS (HEX WORDS) FOR
                          DCP REC
              20   1 N    RESERVED
SF-ID#    21- 22   2 N    MCS ID NUMBER (MCS AND STATION RECORDS)
              23   1 N    RESERVED
```

SPO LOG RECORD SPO-==

SPO-TP	- 0	1	N	RECORD TYPE
SPO-NX	01- 04	4	N	NEXT QUEUE LINK
SPO-RN	05- 08	4	N	RUN LOG NUMBER
SPO-DT	09- 14	6	N	DATE - MMDDYY
SPO-TM	15- 22	8	N	TIME IN MILLISECONDS
SPO-ST	- 23	1	N	SUB TYPE 0=INPUT, 1=OUTPUT
SPO-US	24- 41	9	A	USERCODE
SPO-XX	- 42	1	N	LOG END FLAG
SPO-CU	43- 45	3	N	CHANNEL/UNIT OF REQUESTOR IF SPO-TP=@B@ MESSAGE LENGTH IN DIGITS OTHERWISE
SPO-JN	46- 47	2	N	JOB NUMBER
SPO-LN	48-199	76	A	MESSAGE TEXT

Table A-1. SPO Log Record

The message text may be preceded by "space bell bell" (hex value of "400707" if it was an error message. This should be eliminated before printing, because it will print as

" ??"

on the printer. These characters are the first three bytes of message text when they occur.

The values for SPO-TP are as follows:

"3" The value of 3 indicates input from a program or card reader, or the response to input local ODT's, remote SPO's, or card reader (?SPO <text>).

"A" The hex value of "A" indicates output for an ODT.

"B" The hex value of "B" indicates input from an ODT.

STATUS RECORD DESCRIPTION SRD===

There is one Status Record description (SRD) for each type of label record which maybe present at the beginning of a tape reel. The SRD serves as a header for the Status Record Field Descriptions (SRFD) which follow it.

SRD	0-13		STATUS RECORD DESCRIPTION HEADER
SRDSRD	0- 3	4 N	OFFSET TO NEXT SRD
SRDMIN	4- 5	2 N	MINIMUM # OF TIMES RECORD MAY APPEAR
SRDMAX	6- 7	2 N	MAXIMUM # OF TIMES RECORD MAY APPEAR
SRDMRL	8-11	4 N	MIN. RECORD LENGTH FOR TAPE READ (DIGITS)
SRD#RF	12-13	2 N	NUMBER OF SRFD'S

STATUS RECORD FIELD DESCRIPTION SRF===

There is one Status Record Field Description (SRFD) for each field declared in a label record. There is a minimum of one SRFD for each label record. A tapemark is considered to be a label record, and has a SRFD with a magic number of @01@. The format of the SRFD depends upon the magic number. However all SRFDs possess a common root. The meaning of each magic number may be found at the start of the labeler segment.

```
 SRFD      0- N           STATUS RECORD FIELD DESCRIPTION
 SRFSRF    0- 3    4 N    OFFSET TO NEXT SRFD
 SRFMG#    4- 5    2 N    MAGIC NUMBER

                          FOR ALL MAGIC NUMBERS EXCEPT @01@ = TAPEMARK

 SRFPOS    6- 9    4 N    POSITION OF FIELD IN LABEL (DIGITS)
 SRF-LN   10-11    2 N    LENGTH OF FIELD IN LABEL (DIGITS OR BITS)
 SRF-TY      12    1 N    TYPE OF DATA DESCRIBED
                          =0 - DIGITS
                          =1 - 8-BIT CHARACTERS
                          >=@C@ BIT FIELD WITH BIT OFFSET FROM
                          LEFT (8 BIT) IN 1 & 2 BITS.
                          E.G. @F@ ::= START WITH BIT 1.
 SRF-AC      13    1 N    ACCESS FLAG
                          =0 - MCP ONLY MAY ACCESS
                          =1 - USER MAY READ BUT NOT CHANGE
                          =3 - USER MAY READ OR WRITE
                          >=8 - ONLY VALID IF MATCHING SYSTEM CODE

                          FOR TYPES 31-60

+SRF-PR      14    1 N    STRING PRESENCE FLAG
                          =0 - NO STRING PRESENT
                          =1 - STRING PRESENT (LENGTH IS SRF-LN)
                          =2 - NO STRING - INITIALIZE TO SPACES
                          =3 - NO STRING -INITIALIZE TO ZEROS
                          =4 - NO STRING -INITIALIZE TO ONE
             15    1 N    FILLER
 SRF-VA    16-n           STRING VALUE IF SRF-PR = 1

                          TYPES 11-30 STOP HERE

                          FOR TYPES 03 AND 04

+SRF-DL   14-17    4 N    DATA STRING LENGTH IN BYTES
```

STATUS RECORD FIELD DESCRIPTION SRF===
(Continued)

TYPE 02 = LABID AND
TYPES 61-80 - TRANSLATION DATA

+SRF-DL	14-17	4	N	LENGTH OF TRANSLATION DATA IN DIGITS
	18-n			TRANSLATION DATA
SRFCOD	18-19	2	N	INTERNAL CODE FOR FIRST TRANSLATION
SRFTRN	20	1	N	EXTERNAL CODE FOR FIRST TRANSLATION
			N	REMAINING TRANSLATION CODE PAIRS

```
               DCP STATION TABLE                                    ST-===

     ST-ID#     0-  1   2 N     ATTACHED MCS NUMBER FOR THIS STATION
     ST-MCS     2-  3   2 N     ASSIGNED MCS NUMBER FOR THIS STATION
     ST-PSN     4-  7   4 N     PHYSICAL STATION NUMBER OF THIS STATION
     ST-MMX     8-  9   2 N     MIX NUMBER OF MCS WITH THIS MCS FILE OPEN
     ST-NDL        10   1 N     :8 TRUE = MYUSE INPUT   <<NOT USED BY MCP>>
                                :4 TRUE = MYUSE OUTPUT  <<NOT USED BY MCP>>
                                :2 TRUE = ENABLE INPUT  <<NOT USED BY MCP>>
                                :1 RESERVED
     ST-DC#        11   1 N     DCP NUMBER
     ST-FG1        12   1 N     :8 STATION OUTPUT INHIBITED
                                   (SET WHEN ST-QOT=0)
                                :4 TRUE = STATION ATTACHED
                                :2 RESERVED
                                :1 RESERVED
     ST-QIN        13   1 N     INPUT QUEUE LIMIT <<NOT USED BY MCP>>
     ST-QOT    14- 15   2 N     OUTPUT QUEUE LIMIT (MESSAGE WRITES)
     ST-LSN    16- 19   4 N     LSN OF THIS STATION
     ST-DCP        20   1 N     DCP TABLE INDEX
     ST-SMA    21- 24   4 N     S-MEMORY TABLE ADDRESS
```

STATUS TEMPLATE BLOCK STB===

The status blocks are used by status to determine the label type and translation when a tape is detected by status. There is one STB Header for each label type declared on the system. There is one subfield in the STB Header for each translation type possible for this label type. This field is effectively a line to the first id-field description in the first physical label record for this type. This id-field will in turn be chained to any other id-field descriptions to allow status to quickly determine if this translation of this label matches the label we have read. If it does not, the next translation type will be tried. This process continues until a match is found or the translation types are exhausted. If a matching translation is not found, the process is repeated with all other label types until a match is found or the label types are exhausted. Each status template block contains one or more status record descriptions (SRD). There is one SRD for each physical label type which may appear at the beginning of a tape. For example, the ANSI Label has 4 SRDs; one each for VOL, UVL, HDR, and UHL. Note that it is not necessary to have SRDs for ending labels (EOV, EOF). Each SRD contains one or more Status Record Field descriptions (SRFD). There is one SRFD for each cefined field in a label which status is interested in. For example, there is a SRFD for each label identification field, one for physical tape #, MFID, FID, ETC.

```
          STATUS TEMPLATE BLOCK                                  STB===
                                                            (Continued)

STB       0-23               STATUS TEMPLATE BLOCK
STB-SB    0- 3     4 N       OFFSET TO STB OF NEXT SUBTYPE STB(MATCH)
STB-NX    4- 7     4 N       OFFSET TO STB OF NEXT LABEL TYPE (NO MATCH)
STB-TY    8- 9     2 N       LABEL TYPE CODE
                             CURRENT VALUES ARE
                             01 B3500-B5500 "STANDARD" (" LABEL")
                             02 STANDARD ANSI
                             03 B6700 ANSI
                             04 B3500 ANSI (SIMILAR TO B6700 ANSI)
                             05 CP-MCP ANSI ("BUR")
                             06 OLD MCPV ANSI ("B")
                             07 LABEL1 - INSTALLATION LABEL
                             THIS VALUE CORRESPONDS TO THE VALUE OF IO-LBL
STB-#R   10-11     2 N       NUMBER OF SRD ENTRYS
STBSRD   12-15     4 N       OFFSET TO FIRST SRD
STB-#T   16-17     2 N       NUMBER OF TRANSLATION TYPES POSSIBLE
STB-GR   18-23     6 N       HARDWARE/MODE  RESTRICTIONS
            18     1 N
                             :8 MT7 NRZ ODD NOT PERMITTED
                             :4 MT9 NRZ ODD NOT PERMITTED
                             :2 MT9 PE ODD NOT PERMITTED
                             :1 GRC /PE ODD NOT PERMITTED
            19     1 N       HARDWARE/MODE  RESTRICTIONS (CONT)
                             :8 MT7 NRZ EVEN NOT PERMITTED
                             :4 MT9 NRZ EVEN NOT PERMITTED
                             :2 MT9 PE EVEN NOT PERMITTED
                             :1 GRC /PE EVEN NOT PERMITTED
         20-23     3 N       RESERVED

         Status Block Translation Pointers

STBTRN   24-35    12 N       FIRST STB TRANSLATION POINTER SUBFIELD
STB*XL   24-25     2 N       TRANSLATION TYPE CODE
STB*SR   26-31     6 N       HARDWARE/MODE RESTRICTIONS (SAME AS STB-GR)

         Note that this record starts at ADDR 24.
```

STACK STK===

STACK	0-	n			STACK ENTRY (VARIABLE SIZE)
STKBAK	0-	5	6	N	STACK EXIT RETURN ADDRESS
STKBKA	0-	5	3	A	STACK EXIT RETURN ADDRESS (ALPHA)
STKBOD	0-	16			STACK BODY
STKCAL	16-	17	2	N	MCPCLL CALL SEGMENT NUMBER
STKCLA	16-	17	1	A	MCPCLL CALL SEGMENT NUMBER (ALPHA)
STKINH		14	1	N	TOGGLE INHIBIT
STKIX3	6-	13	8	N	PREVIOUS IX3 STORAGE
STKNTA	20-	25	3	A	MCPCLL OVERLAY ENTRY ADDRESS (ALPHA)
STKNTR	20-	25	6	N	MCPCLL OVERLAY ENTRY ADDRESS
STKOVA	26-	nn		A	MCPCLL STACK PARAMETERS (ALPHA)
STKOVP	26-	nn		N	MCPCLL STACK PARAMETERS
					STACK REDEFINES
STKPAA	16-	nn		A	STACK PARAMETERS (ALPHA)
STKPAR	16-	nn		N	STACK PARAMETERS
STKRTA	18-	19	1	A	MCPOVY RETURN SEGMENT NUMBER (ALPHA)
STKRTN	18-	19	2	N	MCPOVY RETURN SEGMENT NUMBER
STKTOG		15	1	N	TOGGLES

SUBPORT BLOCK SU-===

This is the MCP'S Subport Definition Block. It lives in a type 4 block and contains pointers to the user subport block and to the connected subport when open, or to other subports in the candidates list while waiting open complete. It also contains pointers to the input and output message queues (which are maintained in STCQ blocks) and all of the subport-level attributes. It also contains copies of most of the port-level attributes.

```
SU-IOA   000-099 100 N   STANDARD IOAT FORMAT
SU-MYH   100-133  17 A   MY-HOSTNAME                    [H-NAME]
SU-NAM   134-167  17 A   PORT NAME (TITLE)              [PT-NAM]
SU-GRD   168-179   6 A   GUARDFILE ID                   [PT-GRD]
SU-GFM   180-191   6 A   GUARDFILE FAMILY               [PT-GFM]
SU-STY       192   1 N   SECURITY TYPE                  [PT-STY]
                            1 = PRIVATE
                            2 = PUBLIC
                            4 = GUARDED
                            8 = CONTROLLED
SU-SS        193   1 N   SECURITY USE                  [PT-SUS]
SU-SUB   194-197   4 N   SUBPORT NUMBER (0001 TO MAX-SUBPORTS)
SU-ERR   198-199   2 N   SUBPORT-ERROR
                            0 = NOERROR
                            1 = DISCONNECTED
                            2 = DATALOST (ON CLOSE)
                            3 = NOBUFFER (ON WRITE)
                            4 = NOFILEFOUND (ON OPEN AVAILABLE)
                            5 = UNREACHABLEHOST
SU-STA       200   1 N   SUBPORT-STATE
                            1 = CLOSED
                            2 = OPEN-PENDING
                            3 = OPENED
                            4 = BLOCKED
                            5 = AWATING-HOST
                            6 = DEACTIVATED
                            7 = CLOSE-PENDING
                            8 = CLOSE-BLOCKED
                            9 = DEACTIVATION-PENDING
                            A = ALMOST-OPENED
                            B = SHUTDOWN-IN-PROCESS
                            C = NEVER-OPENED
SU-CHR       201   1 N   :8 PREFERED-CHARACTER-SET = EBCDIC  [PT-CHR]
                         :4 PREFERED-CHARACTER-SET = ASCII
                         :2 ACCEPTABLE-CHARACTER-SET = EBCDIC
                         :1 ACCEPTABLE-CHARACTER-SET = ASCII
SU-CH1       202   1 N   :8 <<AVAILABLE>>
                         :4 <<AVAILABLE>>
                         :2 ACTUAL-CHARACTER-SET = EBCDIC
                         :1 ACTUAL-CHARACTER-SET = ASCII
```

SUBPORT BLOCK

SU-=== (Continued)

SU-CMP	203	1	N	:8 <<AVAILABLE>>	
				:4 RECEIVING-COMPRESSED-DATA	
				:2 SENDING-COMPRESSED-DATA	
				:1 COMPRESSION-POSSIBLE	
SU-MYN	204-403	100	A	MYNAME	[PT-MYN]
SU-USE	404-437	17	A	MY-USERCODE	
SU-JID	438-449	6	A	MY-CODEFILE-NAME	
SU-FAM	450-461	6	A	MY-CODEFILE-FAMILY	
SU-YHN	462-495	17	A	YOUR-HOSTNAME	
SU-YU	496-529	17	A	YOUR-USERCODE	
SU-YOU	530-729	100	A	YOURNAME	
SU-PR	730-737	8	N	ABSOLUTE PORT BLOCK ADDRESS	
SU-LNK	738-745	8	N	ABSOLUTE LINKED SUBPORT BLOCK ADDRESS	
SU-IN	746-751	6	N	INPUT MESSAGE QUEUE LINK (0 = NONE)	
SU-OU	752-757	6	N	OUTPUT MESSAGE QUEUE LINK (0 = NONE)	
SU-MSZ	758-763	6	N	MAX-MESSAGE-TEXT-SIZE	[PT-MSZ]
SU-MSG	764-769	6	N	ACTUAL-MAX-MESSAGE-TEXT-SIZE	
SU-MAX	770-773	4	N	MESSAGE-QUEUE-LIMIT	
SU-INQ	774-777	4	N	INPUT QUEUE COUNT	
SU-RDY	778	1	N	FLOW-STATUS-SENT FLAGS	
	779	1	N	<<AVAILABLE>>	
SU-IN	780-783	4	N	INPUT-EVENT LINK (0000 = NONE, FFFF = LAST)	
SU-CH	784-787	4	N	STATE-EVENT LINK (0000 = NONE, FFFF = LAST)	
SU-RY	788-791	4	N	READY-EVENT LINK (0000 = NONE, FFFF = LAST)	
SU-WR#	792-799	8	N	MESSAGES-SENT	
SU-RD#	800-807	8	N	MESSAGES-RECEIVED	
SU-T1	808-821	14	N	ALLOCATED-TIME-STAMP	
SU-T2	822-835	14	N	OPENED-TIME-STAMP	
SU-CNF	836-843	8	N	CANDIDATE-LIST FORWARD LINK	
SU-CNB	844-851	8	N	CANDIDATE-LIST BACKWARD LINK	
SU-ASO	852-859	8	N	CANDIDATE-LIST ASSOCIATION LINK	
SU-WTG	860-867	8	N	CANDIDATE-LIST WAITING LINK	
SU-FL1	868	1	N	:8 SUBPORT OF MAGIC PLM-PORT	
				:4 "FAKE" MCP-CREATED MATCHING SUBPORT	
				:2 I HAVE MATCHING RESPONSIBILITY	
				:1 WAITING TO SET READY EVENT	
SU-FL2	869	1	N	:8 YOUR-HOSTNAME-WAS-NULL	
				:4 YOURNAME-WAS-NULL	
				:2 SUBPORT IS (WAS) OPEN	
				:1 <<AVAILABLE>>	
SU-BC	870-871	2	N	OPEN TYPE	
SU-BAD	872-875	4	N	MEM BLOCK ADDRESS IN KD	
SU-KC	876-879	4	N	KLUDGE QUEUE COUNT FOR WRITE BEFORE OPEN	
SU-KLN	880-885	6	N	KLUDGE QUEUE LINK FOR WRITE BEFORE OPEN	
SU-LEV	886-889	4	N	DIALOG-PROTOCOL-LEVEL (PLM)	
SU-CUR	890-895	6	N	CURRENT-RECORD-SIZE	
SU-RHN	896-929	17	A	REMOTE HOSTNAME (LIO)	
SU-CLS	930-937	8	N	LINK TO NEXT SUBPORT WAITING CLOSE	

SUBPORT BLOCK SU-===
(Continued)

```
938-999   62 N    <<AVAILABLE>>
```

DCP MCS TABLE SX-===

SX-MCS	0- 11	6	A	MCS NAME
SX-SIZ	12- 19	8	N	SIZE OF MCS BUFFER IN DIGITS
SX-BUF	20- 27	8	N	ABSOLUTE ADDRESS OF MCS BUFFER (MOD 1000)
SX-NXT	28- 35	8	N	ABSOLUTE ADDRESS OF NEXT MCS ENTRY TO BE QUEUED FOR I/O INITIATE
SX-OPC	36- 37	2	N	I/O DESCRIPTOR OPCODE FOR NEXT I/O THIS MCS
SX-VAR	38- 41	4	N	I/O DESCRIPTOR VARIANTS
SX-AAD	42- 47	6	N	I/O DESCRIPTOR A ADDRESS
SX-BAD	48- 53	6	N	I/O DESCRIPTOR B ADDRESS
SX-FUN	54- 55	2	N	I/O DESCRIPTOR FUNCTION CODE
SX-PSN	56- 59	4	N	I/O DESCRIPTOR PHYSICAL STATION NUMBER
SX-LSN	60- 63	4	N	LSN OF CURRENT I/O
SX-IOB	64- 71	8	N	BUFFER RELATIVE ADDRESS OF EXTERNAL BUFFER
SX-CLF	72	1	N	CLOSE IN PROGRESS FLAG
	73- 75	3	N	RESERVED
SX-RJE	76	1	N	RJE STATUS FLAG 0 = TRAP IN MCP SEGMENT DCPB INVALID STATUS 1 = STATION DISABLED ON INITIAL TABLE CONSTRUCTION 2 = TEST AT STATUS IN CASE OF DCP FAILURE 3 = QUE EXECUTION OF THIS RJE MCS, GOOD RESULT HEADER RECEIVED FROM DCP FOR STATION 4 = FILE OPENED 5 = FILE CLOSED 6 = WAITING RJE MCS EXECUTION
SX-TPQ	77	1	N	TOP/BOTTOM QUEUE NEXT I/O
SX-FL1	78	1	N	:8 TRUE = MCS QUEUE ATTACHED :4 TRUE = MCS QUEUE FULL :2 RESERVED :1 RESERVED
SX-FL2	79	1	N	RESERVED
SX-EXB	80- 81	2	N	IOC EXIT CODE FOR THIS I/O
SX-STF	82	1	N	MCS ENABLE STATUS FLAG
SX-TGQ	83- 85	3	N	MAXIMUM NUMBER OF HEADERS ALLOWED IN HDR POOL
SX-MMX	86- 87	2	N	MIX NUMBER OF MCS WITH THIS MCS FILE
SX-ID#	88- 89	2	N	MCS ID NUMBER
SX-DRV	90- 99	10	N	VECTOR TABLE OF DCP'S ASSOCIATED WITH THIS MCS (INDEXED BY DCP INDEX)
SX-QSZ	100-103	4	N	QUEUE SIZE IN BYTES
SX-QBS	104-107	4	N	QUEUE ENTRY TEXT SIZE
SX-QMX	108-111	4	N	MAXIMUM NUMBER OF ENTRIES
SX-QAD	112-119	8	N	ABSOLUTE MEMORY ADDRESS OF Q
SX-QTS	120-123	4	N	TOTAL Q SIZE IN KD
SX-RL#	124-127	4	N	RLOG NUMBER
	128-199	72	N	RESERVED

TIME-SHARING MIX TABLE TMX-==

Deleted as of ASR 6.7

USASII LABEL USA-==

("HDR1" Label Header Record)

The 80 Byte USASII Standard Label begins at relative digit address 56 since the compilers generate a portion of the Burroughs Standard Label which precedes the USASII label. All fields in the USASII Label are in alpha mode.

USALBL	56-215			USASII STANDARD LABEL RECORD
USA-HD	56- 63	4	A	LABEL HEADER / TRAILER IDENTIFIER "HDR1" FOR BEGINNING LABEL "EOR1" FOR END-OF-REEL ENDING LABEL "EOF1" FOR END-OF-FILE ENDING LABEL
	64- 65	1	A	CONSTANT " "
USA-VL	66- 77	6	A	VOLUME ID
	78- 81	2	A	CONSTANT "00"
USA-ID	82- 93	6	A	FILE ID
	94- 97	2	A	CONSTANT "00"
USA-MI	98-109	6	A	MULTI FILE ID
	110-111	1	A	CONSTANT "0"
USA-RL	112-117	3	A	REEL NUMBER
	118-125	4	A	CONSTANT "0001" (FILE SEQUENCE NUMBER)
	126-133	4	A	CONSTANT " " (GENERATION NUMBER)
	134-137	2	A	CONSTANT " " (CYCLE NUMBER)
USA-CD	138-149	6	A	CREATION DATE < YYDDD>
USA-PD	150-161	6	A	PURGE DATE < YYDDD>
	162-163	1	A	CONSTANT " " (ACCESSABILITY)
USA-BC	164-175	6	A	ENDING LABEL BLOCK COUNT
USA-RC	176-189	7	A	ENDING LABEL RECORD COUNT
	190-191	1	A	CONSTANT " "
	192-201	5	A	CONSTANT "BUR " (B 3500 CREATED LABEL)
	202-213	6	A	CONSTANT " " (RESERVED)
USA-SF	214-215	1	A	CONSTANT "1" (USASII STANDARDS FLAG)

USER FILE DESCRIPTION (DISK) USF-==

This disk file contains all the information necessary to validate and grant access to a user. The file is organized as follows:

File contains 4 types of areas
 Usercode entries (106 x 32) described below
 Media list definitions (64 x 53) See Media Definition =?
 Function list definitions (to be specified)
 Resource list definitions (to be specified)

	0-105			
USF-UC	0- 19	10	A	USER CODE
USF-PW	20- 39	10	A	PASSWORD
USF-CN	40- 45	6	N	CHARGE NUMBER
USF-C2	46- 51	6	N	DEFAULT/ALTER CHARGE NUMBER
USF-F1	52	1	N	:8 PASSWORD CHANGE OKAY
				:4 ALLOW DEFAULT CHARGE NUMBER
				:2 OVERRIDE CHARGE NUMBER
				:1 REMOVE ON NULL ACTIVITY
USF-F2	53	1	N	:8 USER PROGRAM USED FOR FURTHER ACCESS SPECF.
				:4 DO NOT REMOVE FLAG
				:2 AVAILABLE
USF-F2				:1 AVAILABLE
USF-US	54- 55	2	N	NUMBER OF USERS ON PROCESSOR #0
	56- 57	2	N	NUMBER OF USERS ON PROCESSOR #1
	58- 59	2	N	NUMBER OF USERS ON PROCESSOR #2
	60- 61	2	N	NUMBER OF USERS ON PROCESSOR #3
USF-MX	62- 65	4	N	VALID MEDIA INDEX (=FFFF IMPLIES ALL VALID)
USF-FX	66- 69	4	N	FUNCTION INDEX
USF-RX	70- 73	4	N	RESOURCE INDEX
USF-ID	74- 79	6	N	USERFL RECORD ID
USF-KY	80- 85	6	N	UNIQUE KEY FOR USERFL RECORD
USF-UN	86-105	10	A	CANDE USER NAME

USER COMBINATION FILE (DISK) USR-==

This file contains the current usercode/password/charge number for all active function requests from IOATs, mixes, etc., keeping track of default codes, function group and resource group keys. The USRTBL entries point to the appropriate entry in this file.

USRCOM	0- 99			USER COMBINATION DISK FILE
USR-UC	0- 19	10	A	USERCODE
USR-PW	20- 39	10	A	PASSWORD
USR-CN	40- 45	6	N	CHARGE NUMBER
USR-FC	46- 53	8	N	FUNCTION GROUP KEY
USR-RS	54- 61	8	N	RESOURCE GROUP KEY
USR-UN	62- 81	10	A	CANDE USER NAME
USR-F1	82	1	N	:8 USER CAN DO LIBMAINT ON OTHER USER'S FILES
				:4 USER CAN DO DIRECT I/O
				:2 RESERVED
				:1 RESERVED
USR-SC	83	1	N	DEFAULT FILE SECURITYCLASS
USR-AP	84	1	N	DEFAULT USER APP MODE
USR-LV	85	1	N	USER CAPABILITY (SPO) LEVEL
USR-F2	86	1	N	:8 USER CAN DO DIALIN
				:4 USER CAN DO COPY TO CTLD
				:2 USER IS CANDE PRIVILEDGED
				:1 RESERVED

USER TABLE DESCRIPTION (DISK) UST-==

This file is used by the security system and is initialized during coldstart when the security card is encountered or after the entire coldstart deck has been read (if there was no security card). There is space for 100 entries/area. There are 13 segments per area, one area for each defined processor on the system.

USRTBL	0-24			USER TABLE DISK DESCRIPTION
UST-LK	0- 3	4	N	LINK TO NEXT ENTRY IN THE CHAIN
UST-UT	4- 5	2	N	USER CODE ENTRY #
UST-TP	6	1	N	TYPE ENTRY
				0 = AVAILABLE
				1 = IOAT ENTRY
				2 = PCR ENTRY
				4 = MIX ENTRY
				8 = MCP TEMP ENTRY
				F = LINKED ENTRY
UST-RF	7-11	5	N	REFERENCE # DEPENDING ON UST-TP, EG MIX #
UST-KY	12-17	6	N	USER FILE KEY
+UST-LS	18-21	4	N	LAST ENTRY IN SEPARATE CHAINS-BASE ENTRY ONLY
+UST-PV	18-21	4	N	PREVIOUS LINK IN CHAIN-OTHER THAN BASE ENTRY
UST-TY	22	1	N	:8 LOG-IN COMBINATION-USED AS DEFAULT
				:4 BEGINUSER COMBINATION-USED AS DEFAULT
				:2 USER COMBINATION LAST INPUT
				:1 AVAILABLE
UST-ST	23	1	N	:8 USERFL MAINTENANCE CAPABILITIES
	24	1	N	AVAILABLE

4 HOW TO READ A PROGRAM DUMP

A program dump is usually taken to find a program problem. This problem can be either a program fault which causes the program to abort or a logic error which causes the program to loop or to perform unexpected logic sequences.

A dump is a snapshot of program memory and shows the contents of memory at a particular point in time. Retracing a program's execution can be done with a dump if the correct program listing is used with it. It is also possible to predict a program's sequence if the necessary information is in memory when a dump is taken.

Program debugging is a subjective tcpic. Like programming, it is never done alike. However, information presented in this section is useful for reading any program dump. All the information will not be applicable for a particular program failure, but having more facts will enable a more complete picture of the program status at the time of the dump.

4.1 OBTAINING A PROGRAM DUMP

A program memory dump can be obtained in several ways. If a program experiences a hardware fault or attempts an illegal action upon a data file, the MCP will terminate the program. If the program had been executed with the MEMDUMP control statement, a program fault will cause the program to abort with a memory dump. If neither MEMDUMP was specified nor the system option TERM was set, the program will be subject to a DS or DP option. A memory dump can be obtained with a DP. The TERM option causes an automatic DS on program faults.

If a program does not abort, a dump can be obtained by a DM or DP keyboard input message. Dumps can also be produced programmatically. In COBOL-74, the construct is CALL SYSTEM DUMP.

Memory dumps are automatically directed to backup disk, and can be identified by IDs of the form $pnnnn. The PM keyboard command is used to initiate the DMPOUT intrinsic to print the dump.

4.2 READING THE DUMP

The following text refers to information in the sample program dump in appendix A.

4.2 READING THE DUMP (Continued)

The printout produced by DMPOUT can be divided into two parts. The first part of the dump shows information which a program does not have access to: IOATs, DFHs, and MIX table entry. The contents of these entries are described in section 5. The second part is the contents of program memory from BASE to LIMIT.

The first pages of the dump give overall information about the program (MIX number, compile date, run date and time, MCP release level, program status, Segment Dictionary, and so on are shown). Following this information, each FIB still open at the time of the dump is printed. IOAT, FIB, and DFH (if applicable) are included.

The Dump Control Segments give the uninterpreted values of the program MIX entry.

The memory contents can be used to determine values at particular memory locations and also to verify the instruction address.

If the dump was produced from a program abort, the MCP gives the reason for and the address of the fault in a message displayed upon the ODT. The message has the following format:

```
-- <Program-ID> <error
condition> <address> <segment #>
```

The value in <address> is the base relative address of the instruction. This value appears in the dump under Run Control Word and is labeled PAR.

If the dump is taken programmatically or through the ODT while the program is running, no error message is displayed, and the PAR value is the address of the next instruction to be executed.

In the memory portion of the dump, the instruction at the address given by PAR should be checked for validity. If the error was MCP generated, (for example, invalid OPENs or I/O errors), the PAR value points at the instruction following the failing instruction. This is also true if the error was a processor interrupt such as an address error or an instruction timeout. For the processor error invalid instruction, PAR points at the failing instruction.

If the failure is an address error, the instruction in question must be decoded. Address errors include

4.2 READING THE DUMP (Continued)

attempting to access outside of BASE and LIMIT, odd address for a UA field, non-mod 4 address for a word field, and branching to an odd address. The address controllers in the instruction indicate whether or not any indexing or indirect addressing is needed. The index registers (IX1, IX2, and IX3) under the heading Reserved Memory, are also found in the memory portion at base relative locations 8, 16, and 24, respectively.

If the program failure is an MCP detected error (for example, file processing errors), the PAR will point at a branch instruction following a Branch Communicate Instruction BCT (OP = 30). This type of failure indicates that a parameter passed to the MCP is incorrect or that one of the other program MCP interface areas is in error. (Refer to the Program Interface, section 2 for BCT formats.)

For a file processing error such as invalid OPEN/CLOSE or invalid READ/WRITE, a FIB is involved. Associated with the FIB are a file header and an IOAT.

Information in the IOAT is not used in most cases of dump reading, but the fields that may be of interest include IO-ST1, IO-ID, IO-PK1, IO-PK2, and IO-RDA. In the disk file header, the information pertaining to the file descriptions for record length and blocking factor are useful (DF-RSZ and DF-RPB).

Several of the FIB fields are useful during program debugging. Some of them are:

FIELD	MEANING
FIBSTA	Describes whether file is OPEN; and if CLOSED, whether previously OPENed.
FIBARB	Depending on buffer technique, current or next record to process.
FIB-WA	Work area location for current record.
FIB-IO	If the file is processed in multiple mode, mode that was most recently used.
FIBRCT	Number of records if any processed before the failure.
FIBBCT	Number of blocks processed.

Aside from the value in determining the general program state, the FIB can be useful in debugging invalid I/O operations (invalid I/O descriptor). This occurs when a program inadvertently destroys the Buffer Status Blocks.

4.2 READING THE DUMP (Continued)

The invalid I/O descriptor can be found by examining the result descriptor (FIBBSW) to which FIB-NB points. For invalid I/O operations, FIBBSW contains a special R/D of the form 9XX2 where the second and third digits specify the specific error in the I/O descriptor, refer to File Buffer Descriptors in section 5.

Invalid programmatic actions detected by the MCP include invalid file OPENs, CLOSEs, READs, WRITEs, and so on. Many failures of this type occur due to inadvertent programmatic destructions of a FIB (for instance, a runaway subscript). Consequently, check the FIB first for such a condition. If the FIB has not been destroyed, several individual fields can be checked. The most valuable FIB field for this type of debugging is FIBSTA, which describes the file status. Most invalid file CLOSEs, READs, and WRITEs occur because the file is not OPEN, while invalid OPENs occur because the file is not CLOSEd.

On occasion other fields can be useful. For invalid READ and WRITE these include:

FIELD	MEANING
FIBMRL	Maximum record size; see FIBARB below.
FIBARB	For READ of variable length records, pointer to current record which may be an invalid size.
FIB-WA	For variable length WRITE analogous to FIBARB above.
FIB-IO	Mode declared at OPEN may be incompatible with I/O request. This often occurs with a COBOL sort when a coding error inadvertently causes the program to fall out of an INPUT PROCEDURE directly into an OUTPUT PROCEDURE.
FIBBLK	2 = variable length; occasionally files are declared variable length inadvertently.
FIBRSW	Work area access records cannot exceed 9999 words.

An invalid OPEN may be observed due to invalid record size, oversize or incompatible disk file declarations, incorrect data communications file declarations, an attempt to use output installations labels without a system installation label definition, or a number of other errors in the FIB. Often the MCP output message on the ODT describes the error.

4.3 STACK OPERATION

A stack is a programmatic method of data storage and retrieval. On B 4000/B 3000/B 2000 series systems, this facility is effected by use of the NTR instruction (OP - 31). The stack is simply a memory area set aside to be used as a stack. The memory area is fixed (once established, cannot change). Depending on the stack location, the upper limit can be either a physical limit (program LIMIT register value) or a logical limit (some predetermined address). In the latter case, there is no absolute protection against stack overflowing (exceeding maximum size).

Figures 7-1, 7-2, and 7-3 illustrate the physical concept of stacks. Figure 7-1 shows an empty stack. Two pointer values (TOP-OF-STACK and CURRENT-ENTRY) are used in stack manipulation. Since the stack is empty, the value of CURRENT-ENTRY is unknown or irrelevant. Figure 7-2 shows the stack after an entry has been placed into it. TOP-OF-STACK now points to the location following the stack entry.

```
                          -------------
               |          |        A    HIGH MEMORY
                                   |
               |          |        |
                                   |
               |          |        |
               |          |        |
               |          |  PROGRAM STACK
               |          |   MEMORY AREA
               |          |        |
               |          |        |
               |          |        |
               |          |        |
               |          |        |
               |          |        |
               |__________|________V____LOW MEMORY
              /
             /
TOP OF STACK
CURRENT ENTRY = n
```

Figure 7-1. A Stack Containing No Entries

4.3 STACK OPERATION (Continued)

```
|              |

|              |

|              |
|              |
|              |
|              |
|              |
|              |
|--------------|  ------> TOP-OF-STACK
|              |
|    STACK     |
|    ENTRY     |
|      1       |
|              |
|______________|  ------> CURRENT ENTRY
```

Figure 7-2. A Stack Containing One Entry

```
|              |

|              |

|              |
|              |
|--------------|  ------> TOP-OF-STACK
|              |
|    STACK     |
|    ENTRY     |
|      1       |
|              |
|--------------|  ------> CURRENT ENTRY
|              |
|    STACK     |
|    ENTRY     |
|      2       |
|              |
|______________|
```

Figure 7-3. A Stack Containing Two Entries

In figure 7-3, another entry has been placed into the stack. CURRENT-ENTRY now points to this entry and TOP-OF-STACK is updated to point at the location where a new stack entry would appear.

If stack entries continue to be created, the stack soon reaches its limit. To prevent this, old or unneeded stack

4.3 STACK OPERATION (Continued)

entries can be discarded, but must be removed from TOP-OF-STACK. With a stack of plates, only the top plate is readily accessible; the bottom plates cannot be reached unless all upper ones have been removed. This action of popping or cutting the stack is accomplished with the EXT (OP = 32) instruction. Cutting the stack is illustrated by reviewing figures 7-1, 7-2, and 7-3 in reverse sequence.

Programmatically, a stack entry is created when an NTR instruction is executed. Since an NTR is a branch-type instruction, but has an eventual return, the current processor state (program address, comparison toggles, and other information) must be saved. The processor state information constitutes a basic stack entry. If parameters are passed with the NTR, these parameters also go into the stack entry. A typical stack entry, appears as shown in figure 7-4.

4.3 STACK OPERATION (Continued)

```
                  |                   |

                  |                   |


                  |                   |
                  |                   |
                  |                   |
TOP ----->|_________________|
OF                |                   |
STACK             |   PARAMETER (n)   |
                  |___________________|
                  |                 . |
                  |                 . |
                  |                 . |
                  |___________________|
                  |                   |
                  |   PARAMETER (1)   |
                  |___________________|  __________________________
                  |                   |                     A
                  |      TOGGLES      |  (1 UN)             |
                  |___________________|                     |
                  |                   |                     |
                  |    1 x 3 VALUE    |  (7 SN)    PROCESSOR STATE
                  |___________________|               16 DIGITS
                  |                   |                     |
                  |  PROGRAM ADDRESS  |                     |
                  |(NEXT INSTRUCTION)|  (6 UN)              V
CURRENT ->|___________________|  __________________________
ENTRY             |                   |

                  |                   |
```

Figure 7-4. Typical Stack Entry

The processor state information requires 16 digits of memory. Program Address is the 6-digit base relative address of the instruction following the NTR. The value of IX3 at the time of the NTR is also saved. The comparison indicator, the overflow indicator, and the ASCII indicator are stored in the Toggles field. Toggles is defined as follows:

Toggles

:8 (ASCII indicator)	1 = ASCII mode, 0 = EBCDIC
:4 (overflow indicator)	1 = Overflow
:21 (Comparison indicator)	1 = High, 2 = Low, 3 = Equal

4.3 STACK OPERATION (Continued)

This layout shows that to access the first parameter in a stack entry, the CURRENT-ENTRY value must be incremented by 16.

The pointers, called CURRENT-ENTRY and TOP-OF-STACK for illustrative purposes, are defined to be fixed locations in a program. The CURRENT-ENTRY pointer is located at BASE:+24:7:SN (IX3 location); TOP-OF-STACK is at BASE:+40:6:UN.

Since IX3 is saved in a stack entry, successive stack entries are linked together by the IX3 value in each stack entry. This is illustrated in the following example.

Assume the following instruction sequence:

```
  L1: NTR  A                                   IX3 : C0004892

      CNST 2 UA = XX                           BASE:+40 : 008000

 L11: DISPLAY . . .
        .
        .
        .

   A: NTR  B

      CNST 6 UN = 012345

  A1: DISPLAY
        .
        .
        .

 EX1: EXT
        .
        .
        .

   B: WRITE

      MVN BASE:+16:Ix3:6    PRINIT:UA:6

      WRITE  PRINIT
 EX2: EXT
```

4.3 STACK OPERATION (Continued)

Upon execution of the NTR A at label L1, the stack would appear as shown in figure 7-5.

```
                              |                  |
                              |                  |
BASE: +40 = 008020 --->       |__________________| PAR = ADDRESS OF A
                              |                  |
                              |        XX        |
                              |__________________|
                              |                  |
                              |     TOGGLES      |
                              |__________________|
                              |                  |
                              |    C0004892      |
                              |__________________|
                              |                  |
                              |  ADDRESS OF L11  |
1 x 3 = C0008000 ----->       |__________________| ADDRESS 8000
```

Figure 7-5. Stack After NTR A

4.3 STACK OPERATION (Continued)

Figure 7-6 shows the stack after NTR B at label A is executed.

```
                          |                       |
                          |                       |
                          |                       |
BASE: + 40 = 008042 ->    |_______________________|
                          |                       |
                          |        .012345        |
                          |_______________________|PAR = ADDRESS OF B
                          |                       |
                          |        TOGGLES        |
                          |_______________________|
                          |                       |
                          |        C0008000       |
                          |_______________________|\
                          |                       |  \
                          |     ADDRESS OF A1     |    \
1 x 3 = C0008020 ---->    |_______________________|      \
                          |                       |        \
                          |          XX           |          \
                          |_______________________|            \
                          |                       |              \
                          |        TOGGLES        |              /
                          |_______________________|            /
                          |                       |          /
                          |        C0004892       |        /
                          |_______________________|      /
                          |                       |    /
                          |     ADDRESS OF L11    |  /
                          |_______________________|/ADDRESS 8000
```

Figure 7-6. Stack After NTR B

4.3 STACK OPERATION (Continued)

After NTRing routine B and going through the EXT at EX2, the stack appears as shown in figure 7-7. Note that the information in the stack is not cleared; rather, the pointers are moved.

```
                                |                     |
                                |                     |
                                |_____________________|
                                |                     |
                                |       .012345       |
                                |_____________________|PAR = ADDRESS OF A1
                                |                     |
                                |       TOGGLES       |
                                |_____________________|
                                |                     |
                                |       C0008000      |
                                |_____________________|
                                |                     |
                                |    ADDRESS OF A1    |
 BASE: +40 = 008020 ->          |_____________________|
                                |                     |
                                |         XX          |
                                |_____________________|
                                |                     |
                                |       TOGGLES       |
                                |_____________________|
                                |                     |
                                |       C0004892      |
                                |_____________________|
                                |                     |
                                |    ADDRESS OF L1    |
 1 x 3 = C0008000 --->          |_____________________|ADDRESS 8000
```

Figure 7.7. Stack After EXT, Routine B

Since IX3 points at the current stack entry, the value cannot be modified if a proper EXT is to be performed. An EXT restores the processor state from the current stack entry. If IX3 does not point at the current stack entry, an EXT will give unpredictable results. Therefore, if IX3 is to be modified in an NTRed routine, the prior value must be SAVEd and then restored before EXT is executed.

4.4 MEMORY DUMP FILE STRUCTURE

The following information pertains to the format of the memory dump backup file created by the MCP. The information is current as of the ASR 6.7 release.

File Naming

Memory dump files are identified by IDs of the form $pnnnn where p = processor number and nnnn = sequential number starting from two. $p0001 is a special ID reserved for MCP usage.

File Layout

A user dump file is a single area file containing unblocked 100-byte records. The number of records per area is calculated according to the following formula:

```
   5 * (Memory assigned to program in KD)
+      (# of non-disk IOATs assigned to program)
+      (# of disk and disk pack files open)
+  5 * (Total size of device alternate blocks in KD)
+  5 * (Total size of address blocks in KD)
+      (# of sequential disk, disk pack or device
        alternates open)
+      (# of spo messages)
+ 35
```

Each non-disk IOAT requires only 100-digits, so 100-digits of filler are added to make one record contain one non-disk IOAT. Space is reserved for an additional 20 non-disk IOATs. Ten records are reserved for the dump control record and the MIX entry. Five records are set aside for a possible Time Sharing Process Stack.

The records indicated above are found in the dump file in the following sequence:

A. Control record 1. Two hundred digits containing miscellaneous information about the program and the dump file. (See definition below.)

B. Control record 2. Two hundred digits containing additional information about the program which is not needed for roll in.

C. MIX. MIX table entry of the program. The size will be 600 digits.

D. FILLER. (1KD digits) Reserved for expansion.

File Layout (Continued)

E. Memory information. Memory image of object program (in whole KD).

F. TSM information. TSM Process Stack or filler if not applicable (1KD).

Note Consult the previous sections for a description of the MIX record.

One or more of the following can be present depending on program requirements:

Device alternate block (N KD).
IOAT (100-digits)
File Header (124-digits)
Buffer (depends on blocking factor and media)

This is followed by:

Disk Address (200 digits)
Possible Address Blocks (1 or 2 KD)

Disk/Disk pack file entry.
IOAT (100-digits)
File Header (100-digits)
Possible Disk Address (200-digits) Only for sequential files
Possible Address Block (1 or 2 KD; depending on the number of areas declared) - Only if present in memory

File Layout (Continued)

Non-Disk/Disk Pack IOAT's (200 digits)

Actual spo log records (each 100 bytes) may be present. The number of records is indicated in the second control record.

Control Record Format

The first record of the dump file is a control record which is used during dump file creation. The information contained in the record pertains to the dump file and to the executing program. The format of control record 1 is:

RELATIVE LOCATION	FIELD NAME	SIZE	FUNCTION
0	H-MCPN	6 UA	MCP name
12	H-ASR#	4 UA	MCP release number
20	H-VERS	6 UN	MCP release date
26	H-CPU	1 UN	CPU type
27	H-MSPD	1 UN	Memory speed
28	H-PROC	2 UN	Processor options (MCPVI only)
30	H-MXID	6 UA	Program ID
42	H-MXMF	6 UA	Multiprogram ID
54	H-MXNO	2 UN	Mix number
56		2 UN	<<Reserved>>
58	H-FLID	6 UA	Dump file ID ($pnnnn)
70	H-PART	8 UN	Snapshot base/limit
78	H-DATE	6 UN	Gregorian date of dump
84	H-TIME	10 UN	Time of dump (milliseconds)
94	H-#SEG	5 UN	# of records in file (EOF pointer)
99	H-PCOR	4 UN	Size of program in KD
103	H-T4#2	2 UN	# of 4KD device alternate blocks
105	H-T4#1	2 UN	# of 2KD device alternate blocks
107	H-DKIO	3 UN	# of disk or disk pack IDATs present
110	H-DKHD	3 UN	# of disk file headers present
113	H-DKAB	3 UN	Size of address blocks in KD
116	H-DKAD	3 UN	# of sequential disk addresses
119	H-DATS	3 UN	# of non-disk IOATs
122	H-RADR	5 UN	Relative disk address of first device alternate block
127	H-OLDB	4 UN	Program base address (KD)
131	H-CORE	7 UN	Memory used by program
138	H-CDT	6 UN	Object program compile date
144	H-CTIM	6 UA	Object program compile time
156	H-SDA	6 UN	Segment dictionary memory address (base relative)

Control Record Format (Continued)

162	H-LEQF	1 UN	Label equate flag
163	H-CMGO	1 UN	0 = Permanent file execute 1 = GO of CMP and GO 2 = Bound program execute
164	H-PSEG	4 UN	# of disk segments in program
168	H-OCOR	4 UN	Original memory for expand memory
172	H-UCOD	10 UA	Usercode of program
192	H-SCTY	1 UN	Default security type
193	H-SCUS	1 UN	Libmaint allowed flag
194		6 UN	<<Reserved>>

The second record of the dump file is a control record which is used by the DMPOUT utility. It contains additional information about the program which is provided to the user in the DMPOUT listing. The format of control record 2 is:

RELATIVE LOCATION	FIELD NAME	SIZE	FUNCTION
0	H2-HNM	17 UA	Hostname
34	H2-MSG	6 UN	Relative disk address of spo log records
40	H2-HMS	4 UN	Number of spo log records (0.100)
44	H2-DIO	6 UN	Relative disk address of first disk or diskpack IOAT
50		150 UN	<Reserved>>

4.5 MCP DUMP FILE STRUCTURE

The following information pertains to the format of the MCP memory dump backup file. The information is current as of the ASR 6.7 release.

File Naming

The MCP memory dump file has an ID of the form $p0001 where p = processor number.

File Layout

The MCP dump file is a single area file containing unblocked 100-byte records as follows:

The first 40 records are reserved.

The next 550 records contains the processor's tables from disk. These include the INSERT area, the mix table, the channel link table, the primary channel

File Layout (Continued)

table, the IOAT's, the disk subsystem table, the EU table, the OCS masks, and the AFTER table.

The next 100 records contain the last 100 records in the spo log file.

The next 10 records are reserved.

The remaining records contain the raw memory dump.