TABLE OF CONTENTS

INTRODUCTION

MCP COMMUNICATES AND STRUCTURES describes many of the areas of MCPII that change considerably from release to release and is a companion document to the MCP product specification 2212 5462, Master Control Program II. It contains four sections and an appendix, covering MCP communicates, declarations, and file attributes. Each section contains its own introduction, and entries, where possible, are indexed alphabetically.

RELATED DOCUMENTATION

Name	Number
B1800/B1700 Software Operational Guide	1068731
Master Control Program II	P.S. 2212 5462

COMMUNICATING WITH THE MCP

In order for a program to request a service of the MCP, it must write a descriptor of the communicate message into the COMMUNICATE.MESSAGE.PTR field of the RUN_STRUCTURE_NUCLEUS. This descriptor may be self-relative or non-self-relative. The MCP treats the two different kinds of descriptors quite differently, breaking the communicates down into four catagories based on the first two bits of the descriptor in this manner:

00 = Run time errors and program interrupts (self-relative)
01 = Standard communicates (non-self-relative)
10 = Invalid communicates
11 = Termination (internal to the MCP)

RUN TIME ERRORS AND PROGRAM INTERRUPTS

The last 6 bits of the type field of the self-relative descriptor in the COMMUNICATES.MESSAGE.POINTER field of the RSN contain the error number. Numbers greater than or equal to 53 are PROGRAM INTERRUPTS. The remainder are RUN TIME ERRORS.

RUN TIME ERRORS

All of these errors cause the program to be discontinued with the reason listed displayed to the operator. Text may be concatenated to the end of the message by supplying a length and base relative address in the communicate message descriptor. Error 29 allows the user program to supply all of the text of the message.

1) PROGRAM POINTER/EVALUATION STACK OVERFLOW

2) CONTROL STACK OVERFLOW

3) NAME/VALUE STACK OVERFLOW

4) REMAP AREA HAS INSUFFICIENT LENGTH

5) INVALID PARAMETER
passed to a procedure

6) INVALID SUBSTRING
or subbit

7) INVALID SUBSCRIPT

8) INVALID RETURN (OF VALUE FROM PROCEDURE)

9) INVALID CASE

10) DIVIDE BY ZERO
could be in a MOD

11) INVALID INDEX

12) MEMORY PARITY ERROR
or READ OUT OF BOUNDS ON B1720

13) INVALID OPERATOR

14) INVALID PARAMETER TO VALUE DESCRIPTOR

15) CONVERT ERROR

16) STACK OVERFLOW

17) UNINITIALIZED DATA ITEM

18) ATTEMPTED TO WRITE OUT OF BOUNDS

19) EXPONENT OVERFLOW

20) EXPONENT UNDERFLOW

21) EXPRESSION OUT OF RANGE

22) SUPERFLUOUS EXIT

23) OUT OF MEMORY SPACE

24) INVALID LINK

25) TYPE ERROR

26) INTEGER OVERFLOW

27) MESSAGE TRANSFER DATA AREA IS NOT PRESENT

28) MESSAGE TRANSFER INVALID DATA TEMPLATE

29) USER SUPPLIED MESSAGE

30) PARAMETER TO DYNAMIC DECLARATION OUT OF RANGE

31) INVALID TRANSLATE

32) INVALID SUBPROGRAM TYPE

33) REFERENCE ASSIGMENT LENGTH MISMATCH

PROGRAM INTERRUPTS

53) read.cass.parity.18

54) s.parity.abort.18

55) out.of.bounds.18

56) shrinkstretch.cleanup

57) halt.now

58) Interpreter segment fault
The address field contains the segment number

59) unknown (high priority interrupt?)

60) put.ready.queue

61) Trace print

62) Code page/segment fault
The first bit of the address field is set to indicate a one Level dictionary. The remainder of the field contains the segment number.

63) Data segment fault
The address field contains the data segment number.

STANDARD COMMUNICATES

A non-self-relative SDL descriptor is used to point to all Standard Communicates. They are in the following format:

F O R M A T

VERB	0 - 11	12
OBJECT	12 - 35	24
ADVERB	36 - 47	12
CT.1	48 - 71	
CT.2	72 - 95	
CT.3	96 - 119	
CT.4	120 - 143	
CT.5	144 - 167	
CT.6	168 - 191	
CT.7	192 - 215	
CT.8	216 - 239	
CT.9	240 - 263	
CT.10	264 - 297	
CT.11	298 - 321	
CT.12	322 - 345	
CT.13	346 - 369	
CT.14	370 - 393	
CT.15	394 - 417	

NOTE: ALL COMMUNICATES RETURN A VALUE OF @000000000000@ OR @000018000000@ IN THE RS.REINSTATE.MSG.PTR UNLESS OTHERWISE SPECIFIED.

CI.VERB 00

CT.VERB 00
ILLEGAL COMMUNICATE

READ = CI.VERB 01

READ (MICRO MCP)

CT.VERB 01
CT.OBJECT FILE.NUMBER
CT.ADVERB BIT
0 REPORT & RETURN TO USER ON EOF
1 REPORT & RETURN TO USER ON PARITY
2 REPORT & RETURN TO USER ON INCOMPLETE I/O
3 LENGTH ADDRESS PAIR IS PRESENT FOR RESULT MASK FIELD
4-6 -
7 STACKERS--STACKER # IS IN CT.3
8-11 -
CT.1 LOGICAL RECORD BIT LENGTH
CT.2 LOGICAL RECORD BASE RELATIVE BIT ADDRESS
CT.3 RANDOM FILE ACTUAL BINARY DISK KEY
(RECORD NUMBER INSERTED BY MCP FOR SERIAL FILES)
OR
LENGTH OF KEY FOR REMOTE FILES AND PORT FILES
CT.4 ADDRESS OF KEY FOR REMOTE FILES AND PORT FILES ONLY
CT.5 LENGTH IN BITS OF RESULT MASK
CT.6 BASE RELATIVE ADDRESS OF RESULT MASK FIELD
REINSTATE.MSG.PTR VALUES
0 GOOD READ
1 END OF FILE
2 I/O ERROR
3 INCOMPLETE I/O
4 IMPOSSIBLE SEARCH (RPG SEARCH OP)

WRITE = CT.VERB 02

WRITE (MICRO MCP)

```
CT.VERB     02
CT.OBJECT   FILE.NUMBER
CT.ADVERB   BIT
            0       REPORT & RETURN TO USER ON EOF
            1       REPORT & RETURN TO USER ON PARITY
            2       REPORT & RETURN TO USER ON INCOMPLETE I/O
            3       LENGTH ADDRESS PAIR IS PRESENT FOR RESULT MASK
                    FIELD
            4       BEFORE/AFTER VARIANT FOR PRINTER FILES
                    0 = BEFORE
                    1 = AFTER
            5       CT.7 CONTAINS MORE ADVERBS
            6       QUEUE FILES: WRITE TO FRONT OF
                    QUEUE ("STACK").
            7       STACKERS--STACKER # IS IN CT.3
            8-11    PRINTER SPACING  (4 BIT VALUE)
                    0  NO PAPER ADVANCE
                    1  SKIP TO CHANNEL 1
                    2  SKIP TO CHANNEL 2
                    3  SKIP TO CHANNEL 3
                    4  SKIP TO CHANNEL 4
                    5  SKIP TO CHANNEL 5
                    6  SKIP TO CHANNEL 6
                    7  SKIP TO CHANNEL 7
                    8  SKIP TO CHANNEL 8
                    9  SKIP TO CHANNEL 9
                    A  SKIP TO CHANNEL 10
                    B  SKIP TO CHANNEL 11
                    C  SKIP TO CHANNEL 12
                    D  TOP OF PAGE (VALID ONLY WITH LINAGE
                       COUNTER)
                    E  SINGLE SPACE
                    F  DOUBLE SPACE
CT.1        LOGICAL RECORD BIT LENGTH
CT.2        LOGICAL RECORD BASE RELATIVE BIT ADDRESS
CT.3        RANDOM FILE ACTUAL BINARY DISK KEY
            (RECORD NUMBER INSERTED BY MCP FOR SERIAL FILES)
            OR
            LENGTH OF KEY FOR REMOTE FILES AND PORT FILES
CT.4        ADDRESS OF KEY FOR REMOTE FILES AND PORT FILES ONLY
CT.5        LENGTH IN BITS OF RESULT MASK
CT.6        BASE RELATIVE ADDRESS OF RESULT MASK FIELD
```

```
  CT.7          EXTENSION OF CT.ADVERB
                BIT
                0       DYNAMIC CHANGE OF LINAGE COUNTER VARIABLES
                1       EXTENDED COMMUNICATE:  WRITE CAT POSITION
                2       NEW COMPILER TYPE - ANSI.74 FOR NOW
                3-23    -
  CT.8          8 BITS PAGESIZE
                8 BITS UPPER MARGIN
                8 BITS FOOTING
  CT.9          8 BITS LOWER MARGIN
                8 BITS NOT USED
                8 BITS SPACING FOR POSITION

  REINSTATE.MSG.PTR VALUES
                0     GOOD WRITE
                1     END OF FILE
                2     I/O ERROR
                3     INCOMPLETE I/O
```

SEEK = CT.VERB 03

```
  SEEK (MICRO MCP)
  ----
  CT.VERB       03
  CT.OBJECT     FILE.NUMBER
  CT.ADVERB     -
  CT.1          -
  CT.2          -
  CT.3          RANDOM FILE ACTUAL BINARY DISK KEY
```

SORTER CONTROL = CT.VERB 04

SORTER CONTROL

CT.VERB	04
CT.OBJECT	FILE.NUMBER
CT.ADVERB	BIT
	0-4 -
	5 TRANSFER
	6 POCKET SELECT
	7 STOP-FLOW
	8 BATCH-COUNT
	9 POCKET LIGHT
	10 -
	11 ENDORSE
CT.1	POCKET NUMBER
CT.2	BASE RELATIVE TRANSFER ADDRESS
CT.3	BIT LENGTH OF TRANSFERRED DATA

SORTER READ = CT.VERB 05

SORTER READ (MICRO MCP)

CT.VERB	05
CT.OBJECT	FILE.NUMBER
CT.ADVERB	-
CT.1	READ AREA BIT LENGTH
CT.2	READ AREA BASE RELATIVE BIT ADDRESS

OPEN (DM) = CI.VERB 06

```
OPEN (DM)
---------
CT.VERB       06
CT.OBJECT     #PATH DICTIONARY ENTRIES (DYNAMIC OPEN BIT 1 OF
               CT.ADVERB=1)
CT.ADVERB     BIT
              0      INCLUDES PACKID OF DICTIONARY
              1      DYNAMIC OPEN (DMINQ)
              2      DM.STATUS FORMAT
                     0=BINARY
                     1=4-BIT DECIMAL
              3      ON EXCEPTION
              4      UPDATE
              5      REORGANIZATION (REORG ONLY)
              6      INCLUDES LOGICAL DATA BASE NAME
              7-11   -
CT.1          DM.STATUS REGISTER BIT LENGTH
CT.2          DM.STATUS REGISTER BASE RELATIVE BIT ADDRESS
CT.3          DATA BASE NAME BASE RELATIVE BIT ADDRESS
CT.4          DATA BASE NAME BIT LENGTH
CT.5          PACKID BASE RELATIVE BIT ADDRESS (BIT 0 OF
                CT.ADVERB = 1)
CT.6          PACKID BIT LENGTH (BIT 0 OF CT.ADVERB=1)
CT.7          LOGICAL DATA BASE NAME BASE RELATIVE BIT ADDRESS
               (BIT 6 OF CT.ADVERB=1)
CT.8          LOGICAL DATA BASE NAME BIT LENGTH (BIT 6 OF
               CT.ADVERB=1)
CT.9 BIT(36)  ABSOLUTE DISK ADDRESS OF PATH DICTIONARY (DYNAMIC OPEN
               BIT 1 OF CT.ADVERB=1)
```

CLOSE (DM) = CI.VERB 07

CLOSE (DM)

CT.VERB	07	
CT.OBJECT	-	
CT.ADVERB	BIT	
	0-1	-
	2	DM.STATUS FORMAT
		0=BINARY
		1=4-BIT DECIMAL
	3	ON EXCEPTION
	4-11	-
CT.1	DM.STATUS REGISTER BIT LENGTH	
CT.2	DM.STATUS REGISTER BASE RELATIVE BIT ADDRESS	

OPEN = CI.VERB 08

OPEN

CT.VERB	08	
CT.OBJECT	FILE.NUMBER	
CT.ADVERB	BIT	
	0	INPUT
	1	OUTPUT
	2	NEW FILE
	3	PUNCH
	4	PRINT
	5	NO REWIND/INTERPRET (DATA RECORDERS)
	6	REVERSE/POCKET (CARD PUNCH)
	7	LOCK
	8	LOCKOUT
	9	REPORT FILE MISSING
	10	REPORT FILE LOCKED
	11	OVERRIDE NAMING CONVENTION AND SECURITY
REINSTATE.MSG.PTR	VALUES	
	0	GOOD OPEN
	1	FILE NOT PRESENT (INPUT DISK)
		PACK NOT PRESENT (OUTPUT DISK)
		NO MORE FILES ON MULTI-FILE REEL (TAPE)
	2	FILE LOCKED (DISK FILES ONLY)

CLOSE = CT.VERB 09

CLOSE

CT.VERB	09	
CT.OBJECT	FILE.NUMBER	
CT.ADVERB	BIT	
	0	REEL
	1	RELEASE
	2	PURGE
	3	REMOVE
	4	CRUNCH
	5	NO REWIND
	6	OVERRIDE NAME CONVENTION AND SECURITY
	7	LOCK
	8	IF NOT CLOSED
	9	ROLLOUT
	10	AUDIT SWITCH
	11	CT.1 CONTAINS MORE TYPES
CT.1	BIT	
	0	UNAVAIL TO ME (COBOL74 LOCK)
	2-20	RESERVED
	21	TERMINATE (MCP INTERNAL)
	22	BACKUP EXTENSION (MCP INTERNAL)
	23	PORT FILE SUBPORT INDEX PRESENT IN CT.2
CT.2	PORT FILE SUBPORT INDEX, 1-RELATIVE (0=ALL)	

POSITION (MICRO MCP (BACKUP FILES ONLY)) = CT.VERB 10

POSITION (MICRO MCP (BACKUP FILES ONLY))

CT.VERB	10	
CT.OBJECT	FILE.NUMBER	
CT.ADVERB	BIT	
	0	REPORT & RETURN TO USER ON EOF
	1	REPORT & RETURN TO USER ON PARITY
	2	REPORT & RETURN TO USER ON INCOMPLETE I/O
	3-7	
	8	POSITION TO END OF FILE
	9	CT.1 CONTAINS PRINTER CHANNEL NUMBER
	10	CT.1 CONTAINS RECORD COUNT AS A FIXED NUMBER
	11	CT.1 CONTAINS RECORD NUMBER DESIRED
CT.1	DEFINED BY BITS IN CT.ADVERB	
REINSTATE.MSG.PTR	VALUES	
	0	GOOD POSITION
	1	END OF FILE (OR END OF PAGE ON PRINTER)
	2	I/O ERROR
	3	INCOMPLETE I/O

ACCESS FILE PARAMETER BLOCK (FPB) = CT.VERB 11

ACCESS FILE PARAMETER BLOCK (FPB)

CT.VERB	11	
CT.OBJECT	FILE.NUMBER	
CT.ADVERB	BIT	
	0-9	-
	10	0=RESTORE NAME
		1=DO NOT RESTORE NAME
	11	0=READ
		1=WRITE
CT.1	RECEIVING FIELD BIT LENGTH	
CT.2	RECEIVING FIELD BASE RELATIVE BIT ADDRESS	

ACCESS FILE INFORMATION BLOCK (FIB) - CI.VERB 12

ACCESS FILE INFORMATION BLOCK (FIB)

```
CT.VERB      12
CT.OBJECT    FILE.NUMBER
CT.ADVERB    BIT
             0-10   -
             11     FORMAT
                    0=CHARACTER
                    1=BINARY
CT.1         RECEIVING FIELD BIT LENGTH
CT.2         RECEIVING FIELD BASE RELATIVE BIT ADDRESS
```

DATA OVERLAY - CI.VERB 13

DATA OVERLAY

```
CT.VERB      13
CT.OBJECT    BASE RELATIVE BIT ADDRESS OF 76 BIT FIELD IN FORMAT OF :
             4 BITS    -
             24 BITS   BEGINNING ADDRESS
             24 BITS   ENDING ADDRESS
             24 BITS   RELATIVE DISK ADDRESS
CT.ADVERB    BIT
             0 - 10    -
             11        - 0 --> READ
                         1 --> WRITE
```

ACCESS DISK FILE HEADER (DFH) = CT.VERB 14

@00E@

ACCESS DISK FILE HEADER (DFH)

CT.VERB	14
CT.OBJECT	BASE RELATIVE ADDRESS OF 30 CHARACTER FILE IDENTIFIER : PACK.ID CAT MFID CAT FID
CT.ADVERB	BIT 0-4 - 5 UPDATE USER'S FILE-NAME FIELD TO THE ONE USED BY THE MCP 6 IN.SECURE 7 REPORT WRITE DID NOT OCCUR 8 ACCESS ON.BEHALF.OF 9-11 0=WRITE 1=READ 2=READ & FORMAT IN BINARY 3=READ & FORMAT IN CHARACTERS
CT.1	RECEIVING FIELD BIT LENGTH
CT.2	RECEIVING FIELD BASE RELATIVE BIT ADDRESS
CT.3	BIT LENGTH OF USERCODE/PASSWORD (SHOULD BE 160)
CT.4	BASE-RELATIVE ADDRESS OF USERCODE/PASSWORD
REINSTATE.MSG.PTR	VALUES 0 COMMUNICATE COMPLETE 1 FILE NOT PRESENT AND IF ACCESS ON.BEHALF.OF: 3 NO USERCODE FILE ON DISK 4 INVALID USERCODE/PASSWORD COMBINATION 5 VIOLATES USERCODE NAMING CONVENTIONS 6 SECURITYTYPE ERROR 7 SECURITYUSE ERROR 8 REQUESTED FILE IS AN EXECUTE-ONLY CODE FILE 9 WRITE DID NOT OCCUR BECAUSE DFH.SELF WAS INCORRECT

IF A SECURITY ERROR IS DETECTED ON AN ACCESS.FILE.HEADER ON.BEHALF.OF, A "READ" WILL BE DONE BUT A "WRITE" WILL NOT BE. IN EITHER CASE, THE REINSTATE.MSG.PTR WILL INDICATE THE RESULT.

BINARY & CHARACTER FORMATS

FIELD	BIT	CHARACTER
OPEN_TYPE	(24)	(01)
NUMBER_OF_USERS	(24)	(02)
RECORD_SIZE	(24)	(04)
RECORDS_PER_BLOCK	(24)	(04)
EOF_POINTER	(24)	(08)
SEGMENTS_PER_AREA	(24)	(08)
USERS_OPEN_OUTPUT	(24)	(01)
FILE_TYPE	(24)	(02)
PERMANENT	(24)	(02)
BLOCKS_PER_AREA	(24)	(06)
AREAS_REQUESTED	(24)	(03)
AREA_COUNTER	(24)	(03)
SAVE_FACTOR	(24)	(03)
CREATION_DATE	(24)	(05)
ACCESS-DATE	(24)	(05)
EXPANDED_RECORD_SIZE	X	(05)
MULTI_PACK_FILE	(01)	(01)
SECURITYTYPE	(02)	(01)
SECURITYUSE	(02)	(01)
UPDATE_DATE	(24)	(05)
CREATE_TIME	(24)	(07)
VERSION	(36)	(12)
EXPANDED_FILE_TYPE	X	(03)

FIND/MODIFY/LOCK (DM) = CT.VERB 15

FIND/MODIFY/LOCK (DM)

```
CT.VERB       15
CT.OBJECT     REMAP# (6 BITS), INVOKE# (6 BITS) & STRUCTURE#
               (12 BITS) OF PATH
CT.ADVERB     BIT
              0      RETURN LIST HEADS (REORG ONLY)
              1      RETURN LOGICAL ADDRESS (REORG & DMINQ ONLY)
              2      DM.STATUS FORMAT
                     0=BINARY
                     1=4-BIT DECIMAL
              3      ON EXCEPTION
              4      DON'T READ DATA RECORD (REORG & DMINQ ONLY)
              5      MODIFY/LOCK
              6      RETURN RECORD.TYPE (REORG & DMINQ ONLY)
              7-10   SELECTION EXPRESSION
                     0  NEXT
                     1  PRIOR
                     2  FIRST
                     3  LAST
                     4  NEXT AT
                     5  CURRENT
                     6  AT
                     7  AT ADDRESS (DMINQ ONLY)
                     8  NEXT GEN SEL EXP
                     9  GEN SEL EXP
              11     DATA SET OR MANUAL SUBSET SELECTION EXPRESSION
CT.1          DM.STATUS REGISTER BIT LENGTH
CT.2          DM.STATUS REGISTER BASE RELATIVE BIT ADDRESS
CT.3          DATASET RECORD WORK AREA BIT LENGTH
CT.4          DATASET RECORD WORK AREA BASE RELATIVE BIT ADDRESS
CT.5          SEARCH KEY (CAT OF COMPONENT NAMES) BASE RELATIVE BIT ADR
               (SEL EXP = 4,6,8,9)
CT.6          REMAP#, INVOKE# & STRUCTURE# OF DATA SET
CT.7 BIT(32)  LOGICAL ADDRESS (SEL EXP =7) (DMINQ ONLY)
CT.7 BIT(24)  LENGTH OF POLISH STRING REQUIRED TO EVALUATE GEN.SEL.EXP
               (SEL EXP = 8,9)
CT.8 BIT(24)  BASE RELATIVE ADDRESS OF GEN.SEL.EXP (SEL.EXP = 8,9)
```

STORE (DM) = CT.VERB 16

STORE (DM)

CT.VERB	16
CT.OBJECT	REMAP# (6 BITS), INVOKE# (6 BITS) & STRUCTURE# (12 BITS) (SUBSET IF INSERT)
CT.ADVERB	BIT 0 INSERT 1 RETURN LOGICAL ADDRESS (REORG ONLY) 2 DM.STATUS FORMAT 0=BINARY 1=4-BIT DECIMAL 3 ON EXCEPTION 4 BEGIN TRANSACTION (NOT INSERT) 5 INCLUDES.LIST.HEADS (REORG ONLY) 6 END TRANSACTION (NOT INSERT) 7 NO AUDIT (BEGIN OR END TRANSACTION ONLY) 8 SYNC (END TRANSACTION ONLY) 9 - 10 STORE INDEXES ONLY (REORG ONLY) 11 PSEUDO CREATE (REORG ONLY)
CT.1	DM.STATUS REGISTER BIT LENGTH
CT.2	DM.STATUS REGISTER BASE RELATIVE BIT ADDRESS
CT.3	DATASET RECORD WORK AREA BIT LENGTH (NOT INSERT)
CT.4	DATA SET RECORD WORK AREA BASE RELATIVE BIT ADDRESS (NOT INSERT)
CT.6	REMAP#, INVOKE# & STRUCTURE# OF DATASET (INSERT)

DELETE (DM) = CI.VERB 17

DELETE (DM)

CT.VERB	17
CT.OBJECT	REMAP# (6 BITS), INVOKE# (6 BITS) & STRUCTURE# (12 BITS) (SUBSET IF REMOVE)
CT.ADVERB	BIT 0 REMOVE 1 - 2 DM.STATUS FORMAT 0=BINARY 1=4-BIT DECIMAL 3 ON EXCEPTION 4-11 -
CT.1	DM.STATUS REGISTER BIT LENGTH
CT.2	DM.STATUS REGISTER BASE RELATIVE BIT ADDRESS
CT.3	DATASET RECORD WORK AREA BIT LENGTH (NOT REMOVE)
CT.4	DATASET RECORD WORK AREA BASE RELATIVE BIT ADDRESS (NOT REMOVE)

CREATE/RECREATE (DM) = CT.VERB 18

CREATE/RECREATE (DM)

CT.VERB	18
CT.OBJECT	REMAP# (6 BITS), INVOKE# (6 BITS) & STRUCTURE# (12 BITS)
CT.ADVERB	BIT 0 - 1 RECREATE 2 DM.STATUS FORMAT 0=BINARY 1=4-BIT DECIMAL 3 ON EXCEPTION 4-11 -
CT.1	DM.STATUS REGISTER BIT LENGTH
CT.2	DM.STATUS REGISTER BASE RELATIVE BIT ADDRESS
CT.3	DATASET RECORD WORK AREA BIT LENGTH
CT.4	DATASET RECORD WORK AREA BASE RELATIVE BIT ADDRESS

SWITCH.TAPE.DIRECTION = CT.VERB 19

SWITCH.TAPE.DIRECTION

CT.VERB	19
CT.OBJECT	FILE.NUMBER
CT.ADVERB	BIT 0 0 = SWITCH TAPE DIRECTION ONLY 1 = RESET FIB FIELDS TO BOF 1-7 NOT USED 8-11 0 = READ FORWARD 1 = READ REVERSE 4 = WRITE
REINSTATE.MSG.PTR	VALUES 0 GOOD SWITCH 1 FILE NOT OPEN 2 WRONG DIRECTION OR NOT A TAPE FILE 3 END OF FILE

TERMINATE (STOP RUN) = CT.VERB 20

TERMINATE (STOP RUN)	
CT.VERB	20

FREE (DM) = CT.VERB 21

FREE (DM)	
CT.VERB	21
CT.OBJECT	REMAP# (6 BITS), INVOKE# (6 BITS) & STRUCTURE# (12 BITS)
CT.ADVERB	BIT 0-1 - 2 DM.STATUS FORMAT 0=BINARY 1=4-BIT DECIMAL 3 ON EXCEPTION 4-11 -
CT.1	DM.STATUS REGISTER BIT LENGTH
CT.2	DM.STATUS REGISTER BASE RELATIVE BIT ADDRESS

TIME/DATE/DAY = CI.VERB 22

TIME/DATE/DAY

CT.VERB	22	
CT.OBJECT	BASE RELATIVE BIT ADDRESS OF WHERE TO PUT THE RESULT	
CT.ADVERB	BIT	
	0	1=DATE REQUESTED
	1-2	FORMAT
		0 YY/DDD (JULIAN)
		1 MM/DD/YY
		2 YY/MM/DD
		3 DD/MM/YY
	3-4	REPRESENTATION
		0 BINARY
		1 4-BIT DECIMAL
		2 8-BIT DECIMAL
	5	1=TIME REQUESTED
	6-7	FORMAT
		0 COUNTER
		1 HH:MM:SS.S (24-HOUR CLOCK)
		2 HH:MM:SS.S TT (12-HOUR CLOCK, TT=AM/PM)
	8-9	REPRESENTATION
		0 BINARY
		1 4-BIT DECIMAL
		2 8-BIT DECIMAL
	10	1=TODAYS.NAME REQUESTED
	11	-

NOTE : TODAYS.NAME RETURNS 9 CHARACTERS LEFT JUSTIFIED

FORMAT	BINARY	4-BIT DECIMAL	8-BIT DECIMAL
YY/DDD (JULIAN)	7+9=16	8+12=20	16+24=40
MM/DD/YY	4+5+7=16	8+8+8=24	16+16+16=48
YY/MM/DD	7+4+5=16	8+8+8=24	16+16+16=48
DD/MM/YY	5+4+7=16	8+8+=24	16+16+16=48
COUNTER	20	24	48
HH:MM:SS.S	5+6+6+4=21	8+8+8+4=28	16+16+16+8=56
HH:MM:SS.S TT	4+6+6+4+16=36	8+8+8+4+16=44	16+16+16+8+16=72
TODAYS.NAME			72 (9 CHAR, LEFT JUST.)

INITIALIZER I/O = CT.VERB 23

INITIALIZER I/O

CT.VERB 23

CT.OBJECT BASE RELATIVE ADDRESS OF
6 BYTE UNIT MNEMONIC
OR
I/O DESCRIPTOR

CT.ADVERB VALUE

0	ASSIGN UNIT TO THIS PROGRAM
1	RELEASE UNIT
2	INVALID
3	LINK IN THE I/O DESCRIPTOR AND INITIATE
4	INVALID

CT.1 IF CT.ADVERB=1 AND CT.1=1 THEN THE MCP WILL "RY"
THE DEVICE AFTER RELEASE

REINSTATE.MSG.PTR VALUES
IF CT.ADVERB=0 THEN
PORT, CHANNEL AND UNIT OF DEVICE REQUESTED
PORT BIT (3)
CHANNEL BIT (4)
FILLER BIT (1)
UNIT BIT (4)
ALL OTHER CASES
0 GOOD COMMUNICATE
1 DISPATCH TO INVALID PORT OR CHANNEL

WAIT (SNOOZE) = CT.VERB 24

WAIT (SNOOZE)

CT.VERB	24
CT.OBJECT	LENGTH OF TIME IN 10THS OF A SECOND
FUNCTION	PROGRAM IS PUT TO SLEEP FOR SPECIFIED LENGTH OF TIME

ZIP = CT.VERB 25

ZIP

CT.VERB	25
CT.OBJECT	-
CT.ADVERB	-
CT.1	MESSAGE AREA BIT LENGTH
CT.2	MESSAGE AREA BASE RELATIVE BIT ADDRESS
REINSTATE.MSG.PTR	VALUES
0	NO ERRORS IN ZIP TEXT
1	ZIPPED INVALID CONTROL CARD

ACCEPT = CT.VERB 26

ACCEPT

CT.VERB	26
CT.OBJECT	-
CT.ADVERB	BIT
0	RETURN IF NO MESSAGE
1-11	-
CT.1	MESSAGE AREA BIT LENGTH
CT.2	MESSAGE AREA BASE RELATIVE BIT ADDRESS
REINSTATE.MSG.PTR	VALUES
0	MESSAGE OF LENGTH ZERO
@FFFFFF@	NO MESSAGE PRESENT
ANY OTHER VALUE	LENGTH OF MESSAGE IN BITS

DISPLAY = CI.VERB 27

```
DISPLAY
-------
CT.VERB     27
CT.OBJECT   -
CT.ADVERB   BIT
            0-10  -
            11    0=CRUNCH BLANKS OUT OF MESSAGE
                  1=PRINT MESSAGE AS IS
CT.1        MESSAGE AREA BIT LENGTH
CT.2        MESSAGE AREA BASE RELATIVE BIT ADDRESS
```

USE/RETURN = CI.VERB 28

```
USE/RETURN
----------
CT.VERB     28
```

SORT HANDLER = CI.VERB 29

```
SORT HANDLER
------------
CT.VERB     29
CT.OBJECT   BASE RELATIVE ADDRESS OF SORT INFORMATION TABLE
CT.ADVERB   BIT (12)
            0 -  SORT.RESTART
            1 -  SORT.DUPCHECK
            2 -  SORT.W1.PID
            3 -  SORT.W2.PID
            4-11 FILLER
CT.1        BASE RELATIVE BIT ADDRESS OF SORT KEY TABLE
CT.2        INPUT FILE.NUMBER OR ADDR OF MERGE.INPUT.TABLE IF MERGE
CT.3        OUTPUT FILE.NUMBER
CT.4        TRANSLATE FILE.NUMBER OR NOT 0
CT.5        -
CT.6        DATA.ADDRESS (DELETE.KEY.TABLE)
CT.7        IF (SORT.W1.PID := W1.PID.FLAG) THEN
               DATA.ADDRESS (W1.PID) ELSE 0
CT.8        IF (SORT.W2.PID := W2.PID.FLAG) THEN
               DATA.ADDRESS (W2.PID) ELSE 0
```

SDL TRACE = CT.VERB

SDL TRACE

CT.VERB	30
CT.OBJECT	TRACE FLAGS

EMULATOR TAPE (MICRO MCP) = CT.VERB 31

EMULATOR TAPE (MICRO MCP)

CT.VERB	31
CT.OBJECT	FILE.NUMBER
CT.ADVERB	BIT 0-2 OP.CODE 0 = READ 1 = WRITE 2 = SPACE 3 = REWIND 4 = TEST 3-8 OP.CODE.VARIANT 3 = REVERSE (READ, SPACE), ERASE (WRITE), TEST.WAIT.READY.NOT.REWIND (TEST) 4 = ONE.RECORD (SPACE), TAPE.MARK (WRITE), TEST.WAIT.NOT.READY (TEST) 5 = ODD.PARITY (READ, SPACE, WRITE) 6 = NOISE (READ, SPACE) 7-8 = NOT USED 9-11 SCHEDULING.VARIANTS 9 = FETCH.RESULT 10 = DONT.WAIT 11 = REPORT AND RETURN ON IO ERROR
CT.1	USER TAPE BUFFER BIT LENGTH
CT.2	USER TAPE BUFFER BASE RELATIVE ADDRESS
CT.3	USER ERROR MASK (BIT SET IMPLIES USER WILL HANDLE THE CORRESPONDING ERROR) BIT 0 (MAY NOT USE)

1	(MAY NOT USE)
2	NOT READY
3	PARITY (NOT ON TEST)
4	ACCESS (NOT ON TEST)
5	TRANSMISSION (ON TEXT ONLY)
6	END.OF.TAPE
7	BEGINNING.OF.TAPE
8	WRITE.LOCK.OUT
9	END.OF.FILE (NOT ON TEST), UNIT.PRESENT (ON TEST)
10	REWINDING
11	TIME.OUT (NOT ON TEST)
12-16	(MAY NOT USE)
17	SHORT.RECORD
18	LONG.RECORD
19	DROPOUT
20	INITIATE.LATE
21	(MAY NOT USE)
22	TRANSMISSION.ERROR.MEC
23	TRANSMISSION.ERROR.MTC

CT.4 BASE RELATIVE ADDRESS OF USER'S 48 BIT RESULT
BIT 0-23 OF RESULT CONTAIN THE RESULT DESCRIPTOR
BIT 24-47 OF RESULT CONTAIN THE ACTUAL LENGTH

REINSTATE.MSG.PTR VALUES

0	= RESULT RETURNED
1	= IO.ERROR
2	= RESULT NOT AVAILABLE

COBOL PROGRAM ABNORMAL END = CT.VERB 32

COBOL PROGRAM ABNORMAL END

CT.VERB	32

SORT EOJ = CT.VERB 33

SORT EOJ

CT.VERB	33
CT.OBJECT	FILE.NUMBER
CT.ADVERB	CLOSE TYPE
CT.1	END-OF-FILE POINTER
CT.2	RECORD SIZE

SEARCH DISK DIRECTORY = CT.VERB 34

@022@

CT.VERB 34

CT.OBJECT FILE NUMBER OF RECEIVING QUEUE (UNUSED IF COUNT ONLY IS REQUESTED) THIS QUEUE FILE MUST BE OPEN IF A LIST OF FILE NAMES IS REQUESTED. THE LIST OF FILE NAMES IS RETURNED TO THIS QUEUE WITH 530-CHARACTER FILE NAME AND HEADER ADDRESS PAIRS PER QUEUE RECORD. THE LAST RECORD MAY CONTAIN FEWER THAN 5 SUCH PAIRS. IT IS THE RECEIVING PROGRAM'S RESPONSIBILITY TO UNBLOCK THE FILE NAMES.

B17(24)~FIXED

CT.ADVERB BIT

0	0 - RETURN FILE NAMES 1 - RETURN FILE COUNT ONLY
1-3	000 - MFID OR MFID/FID 001 - MFID/= 010 - =/MFID OR =/" " 011 - =/=
4-11	UNUSED

CT.1 24 BIT BASE RELATIVE ADDRESS OF 30 CHARACTER NAME FIELD (10 CHARACTER PACK.ID OR UNIT, 10 CHARACTER MFID, 10 CHARACTER FID).

CT.2 24 BIT BASE RELATIVE ADDRESS OF COUNT.
INPUT - COUNT OF FILES TO BE SKIPPED
OUTPUT - COUNT OF FILES FOUND

REINSTATE.MSG.PTR VALUES

0 = GOOD COMMUNICATE
1 = QUEUE FILE OVERFLOW
2 = QUEUE FILE NOT OPEN
3 = NO FILES WERE FOUND
4 = PACK OFF LINE
5 = QUEUE MECHANISM -- NO MEMORY
8 = QUEUE MECHANISM -- NO BUFFERS
9 = QUEUE MECHAISM -- INVALID KEY
10 = QUEUE MECHANISM -- IRRECOVERABLE IO ERROR

FREEZE/THAW RUN STRUCTURE = CT.VERB 35

FREEZE/THAW RUN STRUCTURE

CT.VERB	35
CT.OBJECT	BIT 0 (HIGH ORDER BIT) 0=THAW 1=FREEZE

COMPILE CARD INFORMATION = CI.VERB 36

COMPILE CARD INFORMATION

CT.VERB 36

48 BITS SDL DESCRIPTOR (WHERE TO PUT INFO) IN FORMAT :
8 BITS=FILLER
16 BITS=LENGTH
24 BITS=ADDRESS

RETURNS COMPILE CARD INFO IN FOLLOWING FORMAT :

#CHARS	INFO
30	OBJECT NAME
02	EXECUTE TYPE
10	PACK.NAME OF THE RUNNING PROGRAM
30	INTERPRETER NAME OF THE RUNNING PROGRAM
10	INTRINSIC NAME (FAMILY)
02	PRIORITY
06	SESSION NUMBER
06	JOB NUMBER
20	1ST & 2ND NAMES OF RUNNING PROGRAM
07	CHARGE NUMBER
01	FILLER
36 BITS	DATE AND TIME COMPILED
04 BITS	FILLER
10	USERCODE
10	PASSWORD
04	PARENT JOB NUMBER
20	PARENT QUEUE IDENTIFIER
01	LOG SPO
04	SECONDS BEFORE DECAY
01	PRIVILEGED
02	RESTRICTIONS

DYNAMIC MEMORY BASE = CI.VERB 37

DYNAMIC MEMORY BASE	
CT.VERB	37 VALUE IS RETURNED IN COMMUNICATE MESSAGE POINTER AS SELF RELATIVE DESCRIPTOR

MEMORY DUMP TO DISK = CI.VERB 38

MEMORY DUMP TO DISK	
CT.VERB	38

GET SESSION NUMBER = CI.VERB 39

GET SESSION NUMBER	
CT.VERB	39 SESSION IS PUT INTO RS.REINSTATE.MSG.POINTER

DC.INITIATE.IO = CI.VERB 40

DC.INITIATE.IO	
CT.VERB	40
24 BITS	PORT
24 BITS	CHANNEL
24 BITS	BASE RELATIVE ADDRESS OF I/O DESCRIPTOR

NDL/MACRO COMMUNICATES = CT.VERB 41

NDL/MACRO COMMUNICATES

CT.VERB	41
CT.OBJECT	INDICATES FUNCTION
DESC1	BIT 1-48 MESSAGE AREA 1 (LENGTH & ADDRESS PAIR, BIT (24) EACH)
DESC2	BIT 49-96 MESSAGE AREA 2 (LENGTH & ADDRESS PAIR, BIT (24) EACH)
QUEUE.PTR	BIT 97-106 REMOTE FILE NUMBER, STATION NUMBER, OR QUEUE-FILE-FAMILY ELEMENT NUMBER

FORWARD-CONTROL-MESSAGE

CT.OBJECT	9
DESC1	VIRTUAL LSN
DESC2	MESSAGE AREA

REINSTATE VALUES:

0 SUCCESS
1 DCH, BNA, OR VLSN (VIRTUAL-LSN) TABLE
2 QUEUE FULL
3 NO REMOTE FILE ASSOCIATED WITH THIS VLSN.

THE MCP FINDS THE REMOTE FILE OF HOST/SERVIES THAT IS ASSOCIATED WITH THE SUPPOED VLSN, AND WRITE THE MESSAGE INTO THAT REMOTE FILE QUEUE.

BROADCAST-CLOSE-MESSAGE

CT.OBJECT	10
DESC1	(NOT USED)
DESC2	MESSAGE AREA

REINSTATE VALUES:

0 SUCCESS
1 DCH, BNA, OR VLSN TABLE NOT PRESENT
2 AT LEAST ONE REMOTE FILE QUEUE WAS FULL. THE MCP FORWARDS THE CLOSE MESSAGE TO EACH REMOTE FILE THAT IS ASSOCIATED WITH ANY VLSN IN THE LIST, NOT DUPLICATING MESSAGES TO A REMOTE FILE.

DCWRITE

DCWRITE

CT.OBJECT	11
DESC1	RESULT AREA
DESC2	DC.WRITE MESSAGE
NOTE:	NUMBER AT SUBSTR(DESC2,6,2) IS MESSAGE TYPE
	40=FINISH OPEN
	41=NDL/MACRO PRESENT
	42=ATTACH STATIONS TO REMOTE FILE
	43=DETACH STATIONS FROM REMOTE FILE
	44=RELEASE LINE (NUMBER AT SUBBIT(DESC2,64,12) IS PORT:CHANNEL:ADAPTER)
	PORT=SUBBIT(DESC2,64,3)
	CHANNEL=SUBBIT(DESC2,67,4)
	ADAPTER=SUBBIT(DESC2,72,4)
	45=CONVERT LSN TO STATION NAME OR VICE VERSA
	46=FINISH OPEN (COBOL74)
	47=EXPLICIT CREATE-QUEUE
	48=ENABLE/DISABLE REPLY
	49=EXPLICIT REMOVE-QUEUE

QUICK QUEUE WRITE (REMOTE FILES)

QUICK QUEUE WRITE (REMOTE FILES)

CT.OBJECT	FUNCTION=12
DESC1	MESSAGE HEADER
DESC2	MESSAGE
RMT.FL	REMOTE FILE TO WHICH THE MESSAGE IS DESTINED

(DESC1) CAT (DESC2) is written to the specified remote file input queue, or directly to the application's read buffer if it is in memory, waiting an incoming message and the queue has no other messages in it.

QUEUE WRITE (STATION QUEUE)

QUEUE WRITE (STATION QUEUE)

CT.OBJECT	FUNCTION=13
DESC1	MESSAGE HEADER
DESC2	MESSAGE
ST.NR	STATION NUMBER

(DESC1) CAT (DESC2) is written to the top of the network controller's "STATION.QUEUE". QUEUE.FAMILY element is specified by ST.NR.

QUEUE WRITE (ANY QUEUE)

QUEUE WRITE (ANY QUEUE)

CT.OBJECT	FUNCTION=14
DESC1	MESSAGE HEADER
DESC2	MESSAGE
Q.ELEMENT	Q-FAMILY ELEMENT

(DESC1) CAT (DESC2) is written to the specified FPB.NUM QUEUE FAMILY, ELEMENT #Q.ELEMENT.

GENERAL CT.OBJECT LAYOUT FOR QUEUE WRITES

CT.OBJECT		
BEEN.THRU	BIT(1)	MCP use only; must be set to 0 by program.
FILLER	BIT(2)	
REPORT.Q.FULL	BIT(1)	Functions 13 and 14 only.
WRITE.TO.TOP.OF.Q	BIT(1)	Always done for function=13.
FPB.NUM	BIT(11)	
FUNCTION	BIT(8)	12, 13, or 14

ACCESS USERCODE FILE = CI.VERB 42

Cf SYCOM 71526201

ACCESS USERCODE FILE

CT.VERB 42
DESC 8IT 0-47 DESCRIPTOR TO PARAMETER LIST. (LENGTH & ADDRESS PAIR, BIT (24) EACH)

PARAMETER LIST LAYOUT
MODE BIT (4)

N.B. uc is (surrounded)!

0	SET ALL PARAMETERS IN LIST EXCEPT USERCODE AND PASSWORD. THESE MUST BE SUPPLIED TO FIND CORRECT ENTRY.
1	SET ALL PARAMETERS IN LIST EXCEPT INDEX. INDEX MUST BE SUPPLIED TO FIND ENTRY.
2	SET OVERRIDE. USERCODE MUST BE PRESENT TO FIND ENTRY. THE OVERRIDE FIELD FOR ALL OCCURRENCES OF THIS USERCODE WILL BE SET.
3	SET OVERRIDE. INDEX MUST BE SUPPLIED TO FIND ENTRY. THE OVERRIDE FIELD FOR ALL OCCURRENCES OF THIS USERCODE WILL BE SET.
4	INVALID.
5	INVALID.
6	INITIALIZE ALL OVERRIDE BITS.
7	CHANGE BY USERCODE. ALL ENTRIES FOR A GIVEN USERCODE CAN BE CHANGED WITH ONE COMMUNICATE. USERCODE MUST BE PRESENT. PACK FIELD MUST NOT BE EQUAL TO ZERO TO CHANGE IT. CHARGE NUMBER MUST NOT BE EQUAL TO ZERO TO CHANGE IT. PRIORITY MUST NOT BE EQUAL TO ZERO TO CHANGE IT. CHARGE NUMBER CAN BE CHANGED TO ZERO BY SETTING IT TO A NUMBER LARGER THAN 9999999.
8	INVALID.
9	SET ALL PARAMETERS IN LIST EXCEPT USERCODE AND PASSWORD. ONLY USERCODE HAS TO BE SUPPLIED BECAUSE SEARCH STOPS ON FIRST ENCOUNTER OF GIVEN USERCODE.
10	CHANGE BY INDEX. INDEX MUST BE PRESENT. PRIORITY CAN BE CHANGED BY SETTING FIELD TO NON-ZERO. CHARGE CAN BE CHANGED BY SETTING CHARGE FIELD TO NON-ZERO. PASSWORD CAN BE CHANGED BY SETTING PASSWORD TO NON-ZERO. CHARGE NUMBER CAN BE CHANGED TO ZERO BY SETTING IT TO A NUMBER LARGER THAN 9999999.
11	CLEAR PACK OVERRIDE FIELD FOR ALL OCCURRENCES OF THIS USERCODE. USERCODE MUST BE SUPPLIED.

12 CLEAR PACK OVERRIDE BID FOR ALL OCCURRENCES OF THIS USERCODE. INDEX MUST BE SUPPLIED.

```
INDEX      BIT (10)
USERCODE CHARACTER (10)
           WHEN SET BY PROGRAM (MODE = 0, 2, 4, 5, 7, 8, 9, 11),
             THE USERCODE MAY OR MAY NOT CONTAIN PARENTHESES.
             IF PARENS ARE NOT FOUND, ONLY THE FIRST EIGHT
             USED.
           WHEN SET BY MCP (MODE = 1)
             USERCODE WILL ALWAYS CONTAIN PARENTHESES.
PASSWORD   CHARACTER (10)
PACK NAME CHARACTER (10)
CHARGE #          BIT (24)
PRIORITY          BIT (4)
PRIVILGD          BIT (1)
PUBLIC            BIT (1)
OVERRIDE          BIT (1)
SECURITY LEVEL    BIT (2)
```

REINSTATE.MSG.PTR VALUES

0 NO ERRORS.
1 ERROR ON INPUT: EITHER INDEX IS WRONG OR USERCODE/PASSWORD IS NOT PRESENT.
2 "(SYSTEM)/USERCODE" FILE NOT IN "US" SLOT.

IPC CALL = CI.VERB 43

```
IPC CALL
--------
CT.VERB      43
CT.OBJECT     0 = CALL
              1 = CANCEL
              2 = EXIT PROGRAM (NO EOJ)
CT.ADVERB     BIT
               0     - IF CALL, RETURN ON NO MEMORY
               1-11  - NO USED
CT.1          BASE RELATIVE ADDRESS OF A 30 CHARACTER
              FIELD THAT CONTAINS THE NAME OF THE JOB
              TO BE CALLED OR CANCELLED.
CT.2          NUMBER OF PARAMETERS TO BE PASSED
REINSTATE.MSG.PRT
              0   COMMUNICATE COMPLETED AS REQUESTED
              1   FOR EXIT -
                    PROGRAM DOING COMMUNICATE WAS EXECUTED,
                    NOT CALLED.
```

PROGRAM CALLER = CI.VERB 44

```
PROGRAM CALLER
--------------
CT.VERB       44
48 BITS       SDL DESCRIPTOR
              24 BIT LENGTH OF TEXT
              24 BIT BASE RELATIVE ADDRESS OF TEXT
REINSTATE.MSG.PTR VALUES
               0 NO ERRORS
               1 CONTROL CARD ERROR IN TEXT
```

STACK SIZE CHANGE = CT.VERB 45

STACK SIZE CHANGE	
CT.VERB	45
CT.OBJECT	CODEFILE-RELATIVE DISK ADDRESS OF NEW SPAD

LOAD.DUMP MESSAGE = CT.VERB 46

LOAD.DUMP MESSAGE		
CT.VERB	46	
CT.OBJECT	BASE RELATIVE ADDRESS OF MESSAGE	
CT.ADVERB	BIT	
	0	1=LOADED 0=DUMPED (IF LOAD.DUMP)
	1	0=LOAD.DUMP 1=COPY
	2	0=LIBRARY UPDATE, 1=ERROR (IF COPY)
	3-11	NOT USED

COMPLEX WAIT - CT.VERB 47

COMPLEX WAIT (MICRO MCP)

CT.VERB 47
CT.OBJECT NUMBER OF EVENTS
CT.ADVERB FIRST EVENT TO CHECK (CHECKED IN CIRCULAR FASHION FROM THIS POINT).
CT.1-ETC. BIT ENCODED EVENTS (NUMBER SPECIFIED BY CT.OBJECT MAX=15).

0- 3 EVENT TYPE
4- 7 EVENT PARAM1
8-15 EVENT PARAM2
16-24 EVENT PARAM3

EVENT TYPES:

0 - NULL - PARAM1,2,3 : NOT USED
1 - SPO INPUT PRESENT - PARAM1,2,3 : NOT USED
2 - TIME - PARAM1,2,3 : CONCATENATED BIT 20 FIELD CONTAINING THE LENGTH OF TIME TO WAIT IN 10THS OF A SECOND
3 - READ OK -PARAM1: NOT USED, PARAM2: FILE NUMBER, PARAM3: MEMBER NUMBER IF FILE IS Q-FILE-FAMILY
4 - WRITE OK - PARAM1,2,3: SAME AS READ OK
5 - QUEUE WRITE OCCURRED - PARAM1: NOT USED, PARAM2: FILE NUMBER OF Q-FILE-FAMILY, PARAM3: NOT USED
6 - DATA COMM IO COMPLETE - PARAM1,2,3: NOT USED

REINSTATE.MSG.PTR VALUES
ZERO RELATIVE INDEX TO THE COMMUNICATE EVENT LIST ELEMENT WHICH IS COMPLETE

MESSAGE COUNT = CI.VERB 48

MESSAGE COUNT	
CT.VERB	48
CT.OBJECT	FILE.NUMBER
CT.ADVERB	0 DECIMAL FORMAT RESULTS IF TRUE COBOL ("PIC 999") ELSE BINARY (BIT (24)) 1-11 -
CT.1	RESULT FIELD LENGTH
CT.2	BASE RELATIVE RESULT FIELD ADDRESS
FUNCTION	RETURN THE COUNT OF THE MESSAGES CONTAINED IN THE QUEUE-FILE SPECIFIED. IF THE OBJECT IS A QUEUE-FILE-FAMILY, THE COUNT WILL BE RETURNED AS A LEFT-JUSTIFIED ARRAY OF 24-BIT COUNTS, ONE FOR EACH MEMBER OF THE FAMILY.

CI.VERB 49 = UNUSED

RECOVERY COMPLETE = CI.VERB 50

RECOVERY COMPLETE

CT.VERB	50
CT.OBJECT	DATA BASE GLOBALS BASE RELATIVE BIT ADDRESS

GET.ATTRIBUTES = CI.VERB 51

@ 033 @

GET.ATTRIBUTES

CT.VERB	51
CT.OBJECT	FILE NUMBER
CT.ADVERB	COMMUNICATE LEVEL (MK 8.0 & BEYOND LEVEL=2)
CT.1	TOTAL ATTRIBUTES
CT.2	BASE RELATIVE ADDRESS OF ATTRIBUTE LIST
CT.3	TYPE FOR ANSWER 0 = BINARY 1 = HEX 2 = DECIMAL

NOTE: SEE SECTION IV CSG STANDARD FILE ATTRIBUTES FOR THE LIST OF IMPLEMENTED ATTRIBUTES.
SEE ALSO MCP OPERATORS MANUAL.

CHANGE.ATTRIBUTES = CT.VERB 52

CHANGE.ATTRIBUTES

CT.VERB	52
CT.OBJECT	FILE NUMBER
CT.ADVERB	COMMUNICATE LEVEL (MK 8.0 & BEYOND LEVEL=2)
CT.1	TOTAL ATTRIBUTES
CT.2	BASE RELATIVE ADDRESS OF ATTRIBUTE LIST
CT.3	TYPE FOR NEW DATA 0 = BINARY 1 = HEX 2 = DECIMAL

NOTE: SEE SECTION IV CSG STANDARD FILE ATTRIBUTES FOR THE LIST OF IMPLEMENTED ATTRIBUTES.
SEE ALSO MCP OPERATORS MANUAL.

COBOL74 DATA COMMUNICATIONS (CI.VERB = 53)

```
CT.VERB                         53
CT.OBJECT
  CT.DEST.NUMBER   BIT(12)  % MUST BE ZERO WHEN CT.'IS
                            % RECEIVED BY MCP; USED BY MCP
                            % TO KEEP UP WITH THE ELEMENT
                            % NUMBER OF THE USER'S DESTINATION
                            % TABLE THAT WAS LAST/IS CURRENTLY
                            % BEING SERVICED ON A SEND OR
                            % ENABLE/DISABLE OUTPUT.
  CT.VARIANT       BIT(12)
```

VALUE	MEANING
0	= ACCEPT MESSAGE COUNT
1	= RECEIVE (VARIANTS: MESSAGE [, SEGMENT])
2	= RECEIVE INITIAL INPUT
3	= ENABLE INPUT
4	= ENABLE OUTPUT
5	= SEND (VARIANTS: [ESI,] EMI [, EGI, NONE])
6	= DISABLE INPUT
7	= DISABLE OUTPUT

CT.ADVERB

BIT	MEANING
0-1	- UNUSED
2	- REPORT/RETURN INCOMPLETE I/O (NO DATA)
3-8	- UNUSED
9	- A DISABLED WAS DESTINATION ON A SEND; BIT SET/USED BY MCP ONLY.
10	- PROGRAM HAD AN INVALID DESTINATION ON A SEND; BIT SET/USED BY MCP ONLY.
11	- "BEEN.THRU" BOOLEAN, MEANING SOME MCP INITIALIZATION HAS BEEN DONE; BIT SET/USED BY MCP ONLY.

CT.1

BIT	MEANING
0	- ["SEGMENT" SPECIFIED ON RECEIVE] (ELSE "MESSAGE")
1	- "TERMINAL" SPECIFIED ON ENABLE/DISABLE
2	- CT.4, CT.5 PAIR PRESENT (THEY ARE OPTIONAL.)
3	- 0 = SKIPPING, 1 = SPACING (BOTH USE CT.1 BITS 12-19)

```
      4     -  0 = SKIP/SPACE BEFORE, 1 = AFTER (ON SEND)
      5-11  -  unused
      12-19-   8-BIT SKIPPING/SPACING VALUE (SEND ONLY)
               FOR SKIPPING:
                 1-99 = SKIP TO THIS CHANNEL
                        ("PAGE" = CHANNEL 1)
                        (NO PROVISION FOR "SKIP TO NEXT CHANNEL")
                 0, 100-255 = (INVALID)
               FOR SPACING:
                 <NUMBER OF LINES TO SPACE> (0-255)
      20-23-   4-BIT ENDING INDICATOR VALUE (SEND ONLY)
                 0    = NONE                                  [INVALID]
                 1    = ESI    [END OF SEGMENT INDICATOR---INVALID]
                 2    = EMI    (END OF MESSAGE INDICATOR)
                 3    = EGI    [END OF GROUP   INDICATOR---INVALID]
  CT.2                         % CD BIT LENGTH
  CT.3                         % CD BASE-RELATIVE ADDRESS
  CT.4                         % DATA (OR PASSWORD) BIT LENGTH
  CT.5                         % DATA (OR PASSWORD)
                               % BASE-RELATIVE ADDRESS
  REINSTATE.MSG.PTR VALUES

     0     = GOOD SEND/RECEIVE
     1-2   = UNUSED
     3     = INCOMPLETE I/O (NO DATA)
```

EXPANDED FILE-OPEN = CT.VERB 54

@036@

EXPANDED FILE-OPEN COMMUNICATE

```
CT.VERB      54
CT.OBJECT    FILE NUMBER
CT.ADVERB    BIT
               0      REPORT.FILE.MISSING
               1      REPORT.FILE.LOCKED
               2      REPORT.EXCEPTION (SECURITY ERRORS)
               3      REINSTATE.ASAP (INDICATES A PORT FILE
                      OPEN TYPE OF "RETURN")
               4      SUBPORT INDEX PRESENT IN CT.4 (PORT FILES)
               5-10   -
              11      CT.3 CONTAINS THE FILE NUMBER FOR COBOL
                      MULTI-FILE TAPES
              (THE OPEN TYPE IS TAKEN FROM THE FPB.ADVERB AND
               FPB.EXPANDED.ADVERB FIELDS)
CT.1          LENGTH OF USERCODE/PASSWORD FIELD (IF OPEN.ON.BEHALF.OF)
CT.2          BASE-RELATIVE ADDRESS OF USERCODE/PASSWORD FIELD
              (IF OPEN.ON.BEHALF.OF)
CT.3          FILE NUMBER OF -FPB- TO USE FOR OPEN OF
              MULTI-FILE TAPE FILE.
CT.4          PORT FILE SUBPORT INDEX, 1-RELATIVE (0=ALL)
REINSTATE.MSG.PTR VALUES
               0      SUCCESSFUL OPEN
               1      FILE MISSING (INPUT DISK)
                      PACK NOT PRESENT (OUTPUT DISK)
                      NO MORE FILES ON MULTI-FILE REEL (TAPE)
               2      FILE LOCKED (DISK)
              AND, IF OPEN.ON.BEHALF.OF
               3      NO USERCODE FILE ON DISK (NOT IN NAME TABLE)
               4      INVALID USERCODE/PASSWORD COMBINATION
               5      VIOLATES USERCODE NAMING CONVENTIONS
               6      SECURITYTYPE ERROR
               7      SECURITYUSE ERROR
               8      REQUESTED FILE IS AN EXECUTE-ONLY CODE FILE
               9      FILE WILL BE DISCARDED ON CLOSE LOCK OR CLOSE
                      REMOVE, BUT WAS OPENED
```

THE FILE IS OPENED FOR VALUES OF 0 AND 9, AND NOT OPENED FOR ALL OTHER VALUES.

ACCESS.GLOBALS = CT.VERB 55

@037@

ACCESS GLOBALS

```
CT.VERB      55
CT.OBJECT     0 - CT.3 CONTAINS AN ABSOLUTE MEMORY ADDRESS
              1 - HINTS (CT.3) WILL BE USED AS AN OFFSET
                  INTO HINTS)
              2 - RS.NUCLEUS
              3 - IOAT
              4 - OCH.SCRATCH.MEM
              5 - PACK.INFO.TABLE
              6 - SPO.SW
              7 - DISK COLDSTART VARIABLES
CT.ADVERB FOR CT.OBJECT = 2
              0 = RETURN ALL RUN STRUCTURE NUCLEII
              1 = RETURN RUN STRUCTURE OF THE JOB WHOSE JOB
                  NUMBER IS IN CT.3.  CT.3=0 MEANS OWN JOB
CT.ADVERB FOR CT.OBJECT = 3
              0 = CT.3 CONTAINS AN OFFSET INTO THE IOAT.
                  RETURN FROM OFFSET TO END.
              1 - RETURN THE IOAT ENTRY ASSOCIATED WITH THE
                  FILE NUMBER IN CT.3.
              2 = RETURN THE IOAT ENTRY REQUESTED BY PORT
                  (CHANNEL) UNIT IN CT.3.
CT.ADVERB FOR CT.OBJECT = 7
              0 = RETURN A COMPLETE COPY OF THE DISK COLDSTART
                  VARIABLES.
      1 to 30 = RETURN A COPY OF THE 1-RELATIVE FIELD NUMBER
                  SPECIFIED IN CT.ADVERB FROM THE DISK COLDSTART
                  VARIABLES RECORD.
CT1 AND CT2   BASE RELATIVE SDL DESCRIPTOR WHICH SPECIFIES
              THE RECEIVING FIELD IN THE PROGRAM. (LENGTH &
              ADDRESS PAIR, BIT (24) EACH)
CT.3          DEFINED IN CT.OBJECT AND CT.ADVERB

RS.REINSTATE.MSG.PTR WILL BE ZERO WITH THE FOLLOWING EXCEPTIONS:

CT.OBJECT = 2 and CT.ADVERB = 0   WILL CONTAIN THE NUMBER OF
                                  JOBS IN THE MIX.
CT.OBJECT = 2 and CT.ADVERB = 1   WILL CONTAIN 1 IF THE RE-
                                  QUESTED JOB IS NOT IN THE
                                  MIX.
CT.OBJECT = 3 and CT.ADVERB = 1 or 2
                                  WILL CONTAIN A 1 IF THE RE-
                                  QUESTED IOAT IS NOT THERE OR IF
                                  THE FILE IS NOT OPEN WITH
```

CT.ADVERB = 1.

INDEXED SEQUENTIAL POSITION = CT.VERB 56

INDEXED SEQUENTIAL POSITION

CT.VERB	56	
CT.OBJECT	FILE NUMBER	
CT.ADVERB	BIT	
	0	-
	1	REPORT TO USER ON PARITY
	2	-
	3	RESULT MASK FIELDS PRESENT
	4-5	-
	6-7	RELATIONAL OPERATOR
		0 EQUAL TO
		1 GREATER THAN
		2 NOT LES THAN (> I =)
	8-10	SELECTION CONDITION
		0 NEXT
		1 PRIOR
		2 FIRST
		3 LAST
		4 NEXT AT
		5 CURRENT
		6 AT
		7 RANDOM
	11	-
CT.1	LENGTH OF RESULT MASK	
CT.2	ADDRESS OF RESULT MASK	
CT.3	-	
CT.4	-	
CT.5	STRUCTURE NUMBER	
CT.6	KEY ADDRESS	
CT.7	KEY LENGTH	

INDEXED SEQUENTIAL READ = CT.VERB 57

@039@

INDEXED SEQUENTIAL READ

CT.VERB	57	
CT.OBJECT	FILE NUMBER	
CT.ADVERB	BIT	
	0	REPORT TO USER ON EOF
	1	REPORT TO USER ON PARITY
	2	-
	3	RESULT MASK FIELDS PRESENT
	4-5	-
	6-7	RELATIONAL OPERATOR
		0 EQUAL TO
		1 GREATER THAN
		2 NOT LESS THAN (> I =)
	8-10	SELECTION CONDITION
		0 NEXT
		1 PRIOR
		2 FIRST
		3 LAST
		4 NEXT AT
		5 CURRENT
		6 AT
		7 RANDOM
	11	-
CT.1	LENGTH OF RESULT MASK	
CT.2	ADDRESS OF RESULT MASK	
CT.3	LOGICAL RECORD LENGTH	
CT.4	LOGICAL RECORD ADDRESS	
CT.5	STRUCTURE NUMBER	
CT.6	KEY ADDRESS	
CT.7	-	

INDEXED SEQUENTIAL WRITE - CT.VERB 58

INDEXED SEQUENTIAL WRITE

CT.VERB	58	
CT.OBJECT	FILE NUMBER	
CT.ADVERB	BIT	
	0	REPORT TO USER ON EOF
	1	REPORT TO USER ON PARITY
	2	-
	3	RESULT MASK FIELDS PRESENT
	4-11	-
CT.1	LENGTH OF RESULT MASK	
CT.2	ADDRESS OF RESULT MASK	
CT.3	LOGICAL RECORD LENGTH	
CT.4	LOGICAL RECORD ADDRESS	
CT.5	STRUCTURE NUMBER	
CT.6	KEY ADDRESS	
CT.7	-	

INDEXED SEQUENTIAL REWRITE = CT.VERB 59

INDEXED SEQUENTIAL REWRITE

CT.VERB	59	
CT.OBJECT	FILE NUMBER	
CT.ADVERB	BIT	
	0	-
	1	REPORT TO USER ON PARITY
	2	-
	3	RESULT MASK FIELDS PRESENT
	4-11	-
CT.1	LENGTH OF RESULT MASK	
CT.2	ADDRESS OF RESULT MASK	
CT.3	LOGICAL RECORD LENGTH	
CT.4	LOGICAL RECORD ADDRESS	
CT.5	STRUCTURE NUMBER	
CT.6	KEY ADDRESS	
CT.7	-	

INDEXED SEQUENTIAL DELETE = CT.VERB 60

INDEXED SEQUENTIAL DELETE

CT.VERB	60
CT.OBJECT	FILE NUMBER
CT.ADVERB	BIT
	0 -
	1 REPORT TO USER ON PARITY
	2 -
	3 RESULT MASK FIELDS PRESENT
	4-11 -
CT.1	LENGTH OF RESULT MASK
CT.2	ADDRESS OF RESULT MASK
CT.3	-
CT.4	-
CT.5	STRUCTURE NUMBER
CT.6	KEY ADDRESS
CT.7	-

RELATIVE I/O COMMUNICATE = CI.VERB 61
(MICRO MCP)

RELATIVE I/O COMMUNICATE - START

CT.VERB	61	
CT.OBJECT	FILE NUMBER	
CT.ADVERB	BIT	
	0	REPORT TO USER ON EOF
	1	REPORT AND RETURN TO USER ON PARITY
	2	REPORT AND RETURN TO USER (INCOMPLETE I/O)
	3	RESULT MASK FIELD PRESENT
	4-5	-
	6-7	RELATIONAL OPERATOR
		0 EQUAL TO
		1 GREATER THAN
		2 NOT LESS THAN
	8-11	-
CT.1	LOGICAL RECORD BIT LENGTH	
CT.2	LOGICAL RECORD BASE RELATIVE BIT ADDRESS	
CT.3	ACTUAL BINARY DISK KEY (RELATIVE KEY) SUPPLIED BY USER	
CT.4	-	
CT.5	LENGTH IN BITS OF RESULT MASK FIELD	
CT.6	BASE RELATIVE ADDRESS OF RESULT MASK FIELD	
REINSTATE.MSG.PTR		
	0	GOOD READ
	1	END OF FILE
	2	I/O ERROR
	3	INCOMPLETE I/O

(ADDITIONAL ITEMS FOR FILE STATUS DEFINED IN THE SEQUENTIAL FILES DESIGN SPECIFICATION)

RELATIVE I/O COMMUNICATE WRITE = CI.VERB 62

@03E@

RELATIVE I/O COMMUNICATE - WRITE

CT.VERB	62
CT.OBJECT	FILE NUMBER
CT.ADVERB	BIT
	0 REPORT TO USER ON EOF
	1 REPORT AND RETURN TO USER ON PARITY
	2 REPORT AND RETURN TO USER (INCOMPLETE I/O)
	3 RESULT MASK FIELD PRESENT
	4 ACCESS TYPE
	0 SEQUENTIAL (NEXT)
	1 RANDOM (AT KEY)
	5-11 -
CT.1	LOGICAL RECORD BIT LENGTH
CT.2	LOGICAL RECORD BASE RELATIVE BIT ADDRESS
CT.3	ACTUAL BINARY DISK KEY FOR RANDOM OR DYNAMIC FILES (SUPPLIED BY USER; NOTHING IF IN SEQUENTIAL MODE)
CT.4	-
CT.5	LENGTH IN BITS OF RESULT MASK FIELD
CT.6	BASE RELATIVE ADDRESS OF RESULT MASK FIELD
REINSTATE.MSG.PTR	
	0 GOOD READ
	1 END OF FILE
	2 I/O ERROR
	3 INCOMPLETE I/O

(ADDITIONAL ITEMS FOR FILE STATUS DEFINED IN THE SEQUENTIAL FILES DESIGN SPECIFICATION)

RELATIVE I/O COMMUNICATE REWRITE = CI..VERB 63

RELATIVE I/O COMMUNICATE - REWRITE

```
CT.VERB       63
CT.OBJECT     FILE NUMBER
CT.ADVERB     BIT
              0       REPORT TO USER ON EOF
              1       REPORT AND RETURN TO USER ON PARITY
              2       REPORT AND RETURN TO USER (INCOMPLETE I/O)
              3       RESULT MASK FIELD PRESENT
              4       ACCESS TYPE
                      0  SEQUENTIAL (NEXT)
                      1  RANDOM (AT KEY)
              5-11    -
CT.1          LOGICAL RECORD BIT LENGTH
CT.2          LOGICAL RECORD BASE RELATIVE BIT ADDRESS
CT.3          ACTUAL BINARY DISK KEY FOR RANDOM OR DYNAMIC
              FILES (SUPPLIED BY USER; NOTHING IF IN
              SEQUENTIAL MODE)
CT.4          -
CT.5          LENGTH IN BITS OF RESULT MASK FIELD
CT.6          BASE RELATIVE ADDRESS OF RESULT MASK FIELD
REINSTATE.MSG.PTR
              0       GOOD READ
              1       END OF FILE
              2       I/O ERROR
              3       INCOMPLETE I/O
(ADDITIONAL ITEMS FOR FILE STATUS DEFINED IN THE SEQUENTIAL
 FILES DESIGN SPECIFICATION)
```

THE REWRITE COMMUNICATE WILL BE ESSENTIALLY THE SAME AS THE WRITE, BUT WILL HAVE A DISTINCT MEANING IN LOGICAL I/O

RELATIVE I/O COMMUNICATE DELETE = CT.VERB 64

RELATIVE I/O COMMUNICATE - DELETE

CT.VERB	64
CT.OBJECT	FILE NUMBER
CT.ADVERB	BIT 0 REPORT TO USER ON EOF 1 REPORT AND RETURN TO USER ON PARITY 2 REPORT AND RETURN TO USER (INCOMPLETE I/O) 3 RESULT MASK FIELD PRESENT 4 ACCESS TYPE 0 SEQUENTIAL (NEXT) 1 RANDOM (AT KEY) 5-11 -
CT.1	-
CT.2	-
CT.3	ACTUAL BINARY DISK KEY FOR RANDOM OR DYNAMIC FILES (SUPPLIED BY USER; NOTHING IF IN SEQUENTIAL MODE)
CT.4	-
CT.5	LENGTH IN BITS OF RESULT MASK FIELD
CT.6	BASE RELATIVE ADDRESS OF RESULT MASK FIELD
REINSTATE.MSG.PTR	0 GOOD READ 1 END OF FILE 2 I/O ERROR 3 INCOMPLETE I/O

(ADDITIONAL ITEMS FOR FILE STATUS DEFINED IN THE SEQUENTIAL FILES DESIGN SPECIFICATION)

RELATIVE I/O COMMUNICATE READ = CT.VERB 65

RELATIVE I/O COMMUNICATE - READ

CT.VERB	65
CT.OBJECT	FILE NUMBER
CT.ADVERB	BIT
	0 REPORT TO USER ON EOF
	1 REPORT AND RETURN TO USER ON PARITY
	2 REPORT AND RETURN TO USER (INCOMPLETE I/O)
	3 RESULT MASK FIELD PRESENT
	4 ACCESS TYPE
	0 SEQUENTIAL (NEXT)
	1 RANDOM (AT KEY)
	5-11 -
CT.1	LOGICAL RECORD BIT LENGTH
CT.2	LOGICAL RECORD BASE RELATIVE BIT ADDRESS
CT.3	ACTUAL BINARY DISK KEY FOR RANDOM OR DYNAMIC FILES (SUPPLIED BY USER; NOTHING IF IN SEQUENTIAL MODE)
CT.4	-
CT.5	LENGTH IN BITS OF RESULT MASK FIELD
CT.6	BASE RELATIVE ADDRESS OF RESULT MASK FIELD
REINSTATE.MSG.PTR	
	0 GOOD READ
	1 END OF FILE
	2 I/O ERROR
	3 INCOMPLETE I/O

(ADDITIONAL ITEMS FOR FILE STATUS DEFINED IN THE SEQUENTIAL FILES DESIGN SPECIFICATION)

SEQUENTIAL REWRITE (MICRO MCP) = CI.VERB 66

SEQUENTIAL REWRITE (MMCP)

CT.VERB	66
CT.OBJECT	FILE NUMBER
CT.ADVERB	BIT
	0 REPORT AND RETURN TO USER ON EOF
	1 REPORT AND RETURN TO USER ON PARITY
	2 REPORT AND RETURN TO USER ON INCOMPLETE I/O
	3 LENGTH ADDRESS PART IS PRESENT FOR THE RESULT MASK
	4-11 -
CT.1	LOGICAL RECORD BIT LENGTH
CT.2	LOGICAL RECORD BASE RELATIVE BIT ADDRESS
CT.3	RANDOM FILE ACTUAL BINARY KEY
CT.4	-
CT.5	LENGTH IN BITS OF RESULT MASK
CT.6	BASE RELATIVE ADDRESS OF RESULT MASK FIELD

INDEXED/SEQUENTIAL OPEN = CT.VERB 67

INDEXED/SEQUENTIAL OPEN

CT.VERB	67	
CT.OBJECT	FILE NUMBER	
CT.ADVERB	BIT	
	0	INPUT
	1	OUTPUT
	2	NEW FILE
	3	PUNCH
	4	PRINT
	5	NO REWIND/INTERPRET (DATA RECORDERS)
	6	REVERSE/POCKET (CARD PUNCH)
	7	LOCK
	8	LOCKOUT
	9	REPORT FILE MISSING
	10	REPORT FILE LOCKED
	11	OVERRIDE NAMING CONVENTION AND SECURITY
REINSTATE.MSG.PTR	VALUES	
	0	GOOD OPEN
	1	FILE NOT PRESENT (INPUT DISK) PACK NOT PRESENT (OUTPUT DISK)
	2	FILE LOCKED (DISK FILES ONLY)

ELOG HANDLER - CT.VERB 68

ELOG HANDLER

CT.VERB 68
CT.OBJECT BASE-RELATIVE ADDRESS OF RESULT STATUS FIELD OF THE I/O DESCRIPTOR TO BE LOGGED
CT.ADVERB DEVICE PORT/CHANNEL/UNIT
PORT = 3 BITS
CHANNEL = 4 BITS
FILLER = 1 BIT (MUST BE ZERO)
UNIT = 4 BITS
CT.1 BASE-RELATIVE ADDRESS OF ERROR.INFORMATION.TABLE (FORMAT DESCRIBED BELOW)

THE REINSTATE.MSG.PTR IS NOT USED; THEREFORE, ITS VALUES ARE UNDEFINED>

ERROR.INFORMATION.TABLE FORMAT:

```
DECLARE
    01  ERROR.INFORMATION.TABLE     BIT(368),
        02  LABEL                   CHARACTER(20),
        02  SERIAL.NUMBER           CHARACTER(6),
        02  RETRY.COUNT             BIT(24),
        02  RETRY.RESULT.CODE       BIT(24),
        02  REEL.NUMBER             BIT(24),
        02  BLOCK.NUMBER            BIT(24),
        02  EXTENDED.RESULT         BIT(64);
```

BNA = CI.VERB 69

BNA COMMUNICATE - VERB 69

NETWORK SERVICES VARIANTS (ADVERB = 1)

NSM PRESENT

```
OBJECT     1 (PRESENT)
ADVERB     1 (NS)
CT1        0 (NSM)
CT2        LOCAL NODE-ADDRESS
CT3        NODE MAX MESSAGE SIZE IN BYTES

REINSTATE VALUES
      800 CAT MCP_RELEASE_LEVEL (8 BITS)
          CAT MCP_RELEASE_VERSION (8 BITS)
```

THIS COMMUNICATE NOTIFIES THE MPC THAT THE NETWORK SERVICES MANAGER IS PRESENT AND ENABLES THE MCP TO ROUTE "NW" ODT MESSAGES TO THE NSM.

NWS PRESENT

```
OBJECT       1 (PRESENT)
ADVERB       1 (NS)
CT1          1 (NWS)
CT2          INTERNAL FILE NUMBER OF QUEUE FILE FAMILY FOR NWS OUTPUT
             PORTS (USER INPUT PORTS)
CT3          INTERNAL FILE NUMBER OF QUEUE FILE FAMILY FOR NWS INPUT
             PORTS (USER OUTPUT PORTS)
REINSTATE VALUES
       800 CAT MCP_RELEASE_LEVEL (8 BITS)
           CAT MCP_RELEASE_VERSION (8 BITS)
```

THIS COMMUNICATE NOTIFIES THE MCP THAT THE NETWORK SERVICES PROGRAM IS PRESENT AND INFORMS THE MCP OF THE QUEUE FILE FAMILIES THAT ARE USED TO IMPLEMENT PORT FILES.

HOST SERVICES VARIANTS (ADVERB = 2)

HS/PLM PRESENT

OBJECT 1 (PRESENT)
ADVERB 2 (HS)
CT1 INTERNAL FILE NUMBER OF MCP-TO-PLM CONTROL QUEUE FILE
REINSTATE VALUES
400 CAT MCP_RELEASE_LEVEL (8 BITS)
CAT MCP_RELEASE_VERSION (8 BITS)

THE PLM PRESENT COMMUNICATE INFORMS THE MCP WHAT JOB IS THE BNA PORT LEVEL MANAGER AND OF THAT JOB'S CONTROL QUEUE FILE. THE PLM APPROVES AND/OR DENIES PORT FILE OPENS.

AT RESPONSE

OBJECT 2 (AT RESPONSE)
ADVERB 2 (HS)
CT1 DATA LENGTH OF "MCP TAG"
CT2 DATA ADDRESS OF "MCP TAG"
CT3 DATA LENGTH OF RJE HEADER
CT4 DATA ADDRESS OF RJE HEADER
CT5 DATA LENGTH OF TEXT
CT6 DATA ADDRESS OF TEXT
REINSTATE VALUES
0 SUCCESS
1 FAILURE

THE AT RESPONSE COMMUNICATE IS USED BY HOST SERVICES TO ROUTE RESPONSES OR REPLIES TO "AT" ODT OR ZIP INPUT BACK TO THE PROPER DESTINATION.

PLM FINISH_PORT_OPEN TO MCP

OBJECT 3 FOR USER PORTS, 4 FOR LIO PORTS
ADVERB 2 (HS)
CT1 DATA LENGTH OF PORT OPEN ANSWER MESSAGE
CT2 DATA ADDRESS OF PORT OPEN ANSWER MESSAGE
REINSTATE VALUES
0 SUCCESS
1 JOB NOT WAITING PORT OPEN
2 INVALID JOB NUMBER
3 INVALID USER PROGRAM FILE NUMBER
4 (NOT USED)
5 (NOT USED)
6 (NOT USED)
7 SUBPORT_INDEX = 0 AND OPEN APPROVED
8 FILE NOT EVEN PARTIALLY OPEN
9 ATTRIBUTES PRESENT BUT NWS NOT

THE FINISH PORT OPEN COMMUNICATE INFORMS THE MCP OF THE RESULT

OF A USER PORT FILE OPEN OR A LOGICAL I/O PORT FILE OPEN. THE MESSAGE ALSO TELLS THE MCP WHAT QUEUE ELEMENT OF THE NETWORK SERVICES PROGRAM IS USED FOR THE PORT (SUBPORT).

DIF JOB IF NOT NODIF

```
OBJECT      5 (DIF A JOB)
ADVERB      2 (HS)
CT1         JOB NUMBER
REINSTATE VALUES
            0     SUCCESS
            1     INVALID JOB NUMBER
            2     NO MEM
```

THE DIF COMMUNICATE (DEATH IN FAMILY) REQUESTS THE MCP TO TERMINATE THE SPECIFIED JOB WITH A REASON OF "DEATH IN FAMILY." THE JOB MAY BE IN THE SYSTEM MIX OR SCHEDULED. IF THE JOB'S NODIF ATTRIBUTE IS SET OR THE JOB WAS NOT SPAWNED, THE JOB WILL NOT BE TERMINATED.

ALLOCATE VLSN

```
OBJECT      6 (GET VLSN)
ADVERB      2 (HS)
CT1         INTERNAL FILE NUMBER OF REMOTE FILE
REINSTATE VALUES
            -2    DCH OR BNA NOT PRESENT
            -1    NO MEMORY
             0    NO VLSN AVAILABLE
            <V>   VLSN ASSIGNED
```

THE MCP FINDS AN AVAILABLE VLSN ENTRY, PUTS THE REQUESTOR'S REMOTE FILE NUMBER INTO THAT TABLE ENTRY, ENTERS THE VIRTUAL LSN IN THE REQUESTOR'S REMOTE FILE FIB, AND REINSTATES THE JOB WITH THE VLSN THAT WAS ASSIGNED.

DEALLOCATE VLSN

```
OBJECT      7 (RETURN VLSN)
ADVERB      2 (HS)
CT1         INTERNAL FILE NUMBER OF REMOTE FILE
CT2         VLSN THAT IS BEING RETURNED
REINSTATE VALUES
            0     SUCCESS
            1     INVALID VALUE FOR VLSN
            2     VLSN DOES NOT BELONG TO THIS REMOTE FILE
            3     DCH, BNA, OR VLSN_TABLE NOT PRESENT
```

THE MCP ZEROS OUT THE ENTRY IN THE VLSN TABLE AT <VLSN>, DETACH/CLEARS THE VLSN FROM THE JOB'S REMOTE FILE FIB, AND REINSTATES THE JOB.

POSITION COPY TAPE

```
CT.VERB       70
CT.OBJECT     NUMBER OF FILES TO POSITION
CT.ADVERB     0
CT.1 - CT.8 FILE NUMBERS (ONE FILE NUMBER PER FIELD)
REINSTATE.MSG.PTR VALUES
                0       GOOD POSITION
                BITS    0-7 ARE FLAGS DESIGNATING THE FILES FOR
                        WHICH AN ERROR OCCURED DURING THE POSITIONING
                        OF THE TAPE
                BIT 0 CORRESPONDS TO THE FILE NUMBER IN CT.1
                BIT 1 CORRESPONDS TO THE FILE NUMBER IN CT.2, ETC.
NOTE:  THIS COMMUNICATE TO BE USED BY SYSTEM/COPY ONLY
```

DIAGNOSTIC OPEN = CT.VERB 71

```
CT_VERB       71
CT_OBJECT     FILE NUMBER
CT_ADVERB     BIT
                0-3     OPEN TYPE
                        0 = SINGLE UNIT
                        1 = ALL UNITS SHARING THIS CONTROL
                        2 = SINGLE FILE
                4       0 = EXCLUSIVE USE
                        1 = Other users allowed
                5       0 = Ignore Label  \   Single unit only
                        1 = Verify Label  /
CT_1          Length in bits of Unit Table
CT_2          Base relative address of Unit Table
```

REINSTATE-MSG_PTR Values:

0 - Good communicate
1 - Invalid file number
2 - Invalid communicate
3 - Invalid file
4 - Requested file or unit not found
5 - Requested file or unit(s) could not be assigned
6 - Unit Table too small
7 - Label does not match

DIAGNOSTIC CLOSE = CI.VERB 72

CT_VERB	72	
CT_OBJECT	File number	
CT_ADVERB	BIT	
	0-10	Not used
	11	Terminate
CT_ADVERB	BIT	
	0	0 = No extended result descriptor(s) 1 = Return extended result descriptor(s)
	1	0 = Return control to program after initiating I/O(s) 1 = Wait for completion of I/O(s) before returning control to program
	8-11	Number of descriptors linked to initial descriptor
CT_1	Base relative address of I/O Descriptor	
CT_2	Base relative address of 64-bit extended	

REINSTATE_MSG_PTR Values:

0 - Good communicate
1 - Invalid file number
2 - Invalid communicate
3 - Invalid file
4 - Invalid I/O Descriptor
5 - Violates restricted access
(Named disk file or shared disk unit only)
6 - No descriptors available; request ignored
7 - Unit is in use by a different control
(Tape exchange only)

DECLARATIONS

DISK AVAILABLE

```
    DEFINE
    DISK.AVAILABLE.DECLARATION AS#
DECLARE
    01 DUMMY REMAPS DISK.AVAILABLE BIT(SEG.SIZE),
       02 AVL.FOR.LINK          DSK.ADR,
       02 AVL.BACK.LINK         DSK.ADR,
       02 AVL.SELF              DSK.ADR,
       02 FILLER                BIT(4),
       02 AVL.BLOCK(22),
          03 AVL.ADDRESS        DSK.ADR,
          03 AVL.LENGTH         WORD;#;
 %
 %
 %
```

PACK LABEL

```
DEFINE  PACK_LABEL_SIZE AS #180#; % BYTES
RECORD
01 PACK_LABEL_DECLARATION CHARACTER (PACK_LABEL_SIZE)
,    02 PL.VOL1                  CHAR (4)  %          "VOL1"
,    02 PL.SERIAL.NO             CHAR (6)  %          SERIAL (CAN) NUMBER
,    02 PL.ACCESS.CODE           CHAR (1)  %          ACCESS CODE
,    02 PL.ID                    CHAR (17) %          PACK ID
,         03 PL.NAME             CHAR (10) %
,         03 FILLER              CHAR (7)  %
,    02 PL.SYSTEM.INTERCHANGE    CHAR (2)  %          SYSTEM INTERCHANGE/CODE
                                           %  00 = INTERCHANGE
                                           %  17 = B1700 INTERNAL
                                           %  35 = B6500 INTERNAL
                                           %  ETC, ETC, ETC
,    02 PL.CODE                  CHAR (1)  %          PACK CODE 00 = SCRATCH
,    02 FILLER                   CHAR (6)  %
,    02 PL.OWNER.ID              CHAR (14) %
,    02 PL.TYPE                  CHAR (1)  %          "R" = RESTRICTED PACK
                                           %          "U" = USER PACK
                                           %          "S" = SYSTEM.PACK


,    02 PL.CONTINUE              CHAR (1)  %          CONTINUATION FLAG "C"
,    02 FILLER                   CHAR (26) %
,    02 PL.INT                   CHAR (1)  %
,    02 PL.VOL2                  CHAR (4)  %          "VOL2"
,    02 PL.DATE.INITIALIZED      CHAR (5)  %
,    02 PL.INIT.SYSTEM           CHAR (6)  %          INITIALIZING SYSTEM
,    02 PL.DISK.DIRECTORY        CHAR (8)  %          DIRECTORY ADDRESS
,    02 PL.MASTER.AVAIL          CHAR (8)  %          MASTER AOAILABLE TABLE
,    02 PL.DISK.AVAILABLE        CHAR (8)  %          WORKING AVAILABLE TABLE
,    02 PL.INTEGRITY             CHAR (1)  %          0 = NORMAL
                                           %          1 = RECOVERY REQUIRED
,    02 PL.ERROR.COUNT           CHAR (6)  %
,    02 PL.SECTORS.XD            CHAR (6)  %          REMOVED SECTORS
,    02 PL.TEMP.TABLE            CHAR (8)  %          TEMP TABLE LINK
,    02 PL.PCD                   CHAR (3)  % LAST PORT, CHAN, DRIVE
,    02 PL.ASSIGNED.TO.BPS       CHAR (6)  %    BASE PACK SERIAL NUMBER
,    02 PL.SP.SEC.FLAGS           CHAR (8)  % SPARE.SECTOR.TABLE FOR 225 DSK
,    02 FILLER                    CHAR (23)
;
  %
  %
  %
```

FILE HEADER

```
RECORD
  01 DFH_RECORD                          BIT(580),
    02 DFH_AREA_ADDR_OFFSET              BIT(16),
       %OFFSET INTO THE DFH (IN BITS) FOR THE FIRST AREA ADDRESS
    02 DFH_FILE_TYPE                     BIT(8),
       %TYPE OF FILE DESCRIBED BY THIS HEADER
    02 DFH_SELF                          DSK_ADR,
       %DISK ADDRESS OF THIS HEADER
    02 DFH_NO_USERS                      BIT(8),
       %NUMBER OF USERS WHO HAVE THIS FILE OPENED
    02 DFH_USERS_OPEN_OUT                BIT(4),
       %NUMBER OF USERS WHO HAVE THIS FILE OPENED I/O OR OUTPUT
    02 DFH_OPEN_TYPE                     BIT(4),
       %HOW THIS FILE WAS OPENED
       %      BIT 0 = LOCKOUT
       %      BIT 1 = LOCK
       %      BIT 2 = OUTPUT
       %      BIT 3 = INPUT
    02 DFH_FILE_TYPE_8_0                 BIT(4),
       %PRE-9.0 FILE TYPES
    02 DFH_PERMANENT                     BIT(4),
       %HOW PERMANENT THIS FILE IS. THE VALUES ARE ---
       %  0 = TEMPORARY - WILL BE REMOVED NEXT CLEAR/START
       %  1 = PERMANENT - NORMAL FILES CONTAIN THIS VALUE
       %  2-D          NOT    USED
       %  E = IAD FILE - CANNOT BE MOVED BY SQUASH
       %  F = SYSTEM FILE - CANNOT REMOVE, CHANGE OR SQUASH
    02 DFH_JOB_WAITING_ON_CLOSE          BIT(1),
       %SOMEONE ATTEMPTED TO OPEN THIS FILE BUT COULDN'T BECAUSE
       %IT IS CURRENTLY OPENED LOCK OR THE REQUESTOR WANTS TO OPEN
       %IT LOCK AND ITS IN USE. TELLS CLOSE TO CAUSE ANY JOBS
       %WAITING NO FILE WHEN THIS FILE IS CLOSED.
    02 DFH_NEWFILE                       BIT(1),
       %THIS FILE IS NOT IN THE DIRECTORY YET
    02 FILLER                            BIT(6),
    02 DFH_HDR_SIZE                      BIT(16),
       %TOTAL SIZE OF THEIS HEADER (IN BITS)
    02 DFH_NO_USERS_LOCK                 BIT(4),
       %NUMBER OF USERS WHO HAVE THIS FILE OPENED WITH LOCK
    02 DFH_RECORD_SIZE                   BIT(20),
       %SIZE OF THE RECORDS (IN BITS)
    02 DFH_FILE_LEVEL                    BIT(4),
       % 0 = 8.0 AND EARLIER
       % 1 = 9.0
    02 DFH_RCDS_BLOCK                    BIT(20),
       %NUMBER OF RECORDS PER BLOCK
```

```
02 DFH_BLOCKS_AREA                    WORD,
     %NUMBER OF BLOCKS PER AREA
02 DFH_SEGS_AREA                      WORD,
     %NUMBER OF SEGMENTS OR SECTORS PER AREA
02 DFH_AREAS_RQST                     BIT(12),
     %MAXIMUM NUMBER OF AREAS ALLOWED IN THIS FILE
02 DFH_AREA_CTR                       BIT(12),
     %CURRENT HIGH AREA NUMBER ALLOCATED
02 DFH_EOF_POINTER                    WORD,
     %HIGHEST RECORD NUMBER WRITTEN IN THIS FILE
02 DFH_AUDITED                        BIT(1),
     %DO NOT REINSTATE USER UNTIL I/O COMPLETE
02 DFH_PROTECTION_ATTR                BIT(2),
     %HOW BADLY DOES THE USER WISH TO SAVE THIS FILE
     %IN THE OFF CHANCE OF A CLEAR/START WHILE OPEN?
     % 0 = TEMPORARY
     % 1 = ABNORMALSAVE
     % 2 = SAVE
     % 3 = PROTECTED
02 FILLER
02 DFH_BPS_NO                         BIT(20),
     %SERIAL NUMBER OF THE BASE PACK TO WHICH THIS MULTI-PACK
     %FILE BELONGS
02 FILLER                             BIT(27),
02 DFH_MPF                            BIT(1),
     %THIS IS A MULTI-PACK FILE
02 DFH_UPDATE_DATE                    BIT(16),
     %JULIAN DATE OF THE LAST TIME THIS FILE WAS CLOSED AFTER
     %HAVING BEEN WRITTEN ON. ALSO DATE OF LAST NAME CHANGE.
     %FOR CODE FILES, ITS THE DATE OF THE LAST MODIFY.
02 FILLER                             BIT(4),
02 DFH_CREATE_TIME                    BIT(20),
     %TIME THE FILE WAS OPENED OUTPUT NEW.
02 FILLER                             BIT(32),
02 DFH_SAVE_FACTOR                    BIT(12),
     %NUMBER OF DAYS TO SAVE THIS FILE. NO SIGNIFICANCE.
02 DFH_CREATION_DATE                  BIT(16),
     %JULIAN DATE OF WHEN THIS FILE WAS OPENED OUTPUT NEW.
02 DFH_ACCESS_DATE                    BIT(16),
     %JULIAN DATE OF WHEN THIS FILE WAS LAST OPENED. FOR CODEFILES,
     %DATE LAST EXECUTED OR MODIFIED.
02 FILLER                             BIT(61),
02 DFH_UPDATE_VERSION                 BIT(1),
     %DMS USE ONLY
02 FILLER                             BIT(2),
02 DFH_VERSION
     %TIME AND DATE OF THE LAST CLOSE. I/S AND DMS ONLY.
02 DFH_PROTECTION                     BIT(2),
     % 0 = PUBLIC FILE
     % 1 = PRIVATE FILE
```

```
    02 DFH_PROTECTION_IO                    BIT(2),
         % 0 = ACCESS MAY BE I/O
         % 1 = ACCESS MAY BE INPUT ONLY
         % 2 = ACCESS MAY BE OUTPUT ONLY
    02 DFH_USERS_RANDOM                     BIT(8),
         %NUMBER OF USERS WHO HAVE THIS FILE OPENED RANDOM
    02 FILLER                               BIT(8),
    02 DFH_MINRECSIZE                       BIT(20),
         %MINIMUM NUMBER OF BITS IN EACH LOGICAL RECORD
    02 DFH_MAXRECSIZE                       BIT(20);
         %MAXIMUM NUMBER OF BITS IN EACH LOGICAL RECORD


RECORD
   01 AN_AREA_ADDRESS                       BIT(36)
      02 DFH_UNIT                           BIT(12),
         03 DFH_PC,                         BIT(7)
            04 DFH_PORT                     BIT(3),
            04 DFH_CHAN                     BIT(4),
         03 DFH_SER_NO_FLAG                 BOOLEAN,
         03 DFH_EU                          BIT(4),
         03 DFH_ADDR                        BIT(24);

RECORD
   01 AREA_ADDRS                            BIT(3780)
      02 AREA_ADDR (105)                    AN_AREA_ADDRESS,
      02 FIRST_AREA REMAPS AREA_ADDR AN_AREA_ADDRESS;


 %
 %
```

LABEL SIZE

```
DEFINE LBL_SIZE AS #640#;
 %
DEFINE
     SCRTCH_TYPE AS #"0"#
,    USER_TYPE AS #"1"#%
,    BACKUP_TYPE AS #"2"# %
,    LIBRARY_TYPE AS#"3"# %
,    NOT_ANSI AS #"C"# %
,    BOV  AS #1# %
,    BOF  AS #2# %
,    EOV  AS #3# %
,    EOF AS #4# %
,    PFB  AS #5# %
,    LOST AS #7# %
;       %
RECORD 01 DUMMY STANDARD_LABEL_RECORD  CHAR (80)
,    02 L_LABEL      CHAR (9)  %          " LABEL  0"
,    02 L_MFID       CHAR (7)  %                  "
,    02 L_Z1         CHAR (1)  %          "0"     "
,    02 L_ID         CHAR (7)  %
,    02 L_REEL       CHAR (3)  %
,    02 L_DW         CHAR (5)  %          DATE WRITTEN
,    02 L_CYCLE      CHAR (2)  %          "00"
,    02 L_PID        CHAR (5)  %          PURGE DATE
,    02 L_S          CHAR (1)  %          SENTINNEL (1 = END-OF-REEL)
,    02 L_BC         CHAR (5)  %          BLOCK COUNT
,    02 L_RC         CHAR (7)  %          RECORD COUNT
,    02 L__B         CHAR (1)  %          PRINT BACKUP FLAG


,    02 L_SERIAL     CHAR (5)  %          SERIAL NUMBER
,    02 L_SYSTEM     CHAR (5)  %          CREATING SYSTEM
,    02 L_BUFSIZE    CHAR(8)   %          NEW FORMAT DECIMAL BLOCK SIZE
,      03 L_BSIZE    BIT(24)   %          OLD FORMAT BINARY
,      03 L_RSIZE    BIT(24)   %          OLD FORMAT BINARY
,    02 L_RECSIZE    CHAR(8)   %          NEW FORMAT DECIMAL RECORD SIZE
,    02 L_MODE       CHAR(1)   %          NEW FORMAT RECORDING MODE FOR
                               %          TAPE FILE
RECORD
      01 ANSI_TAPE_RECORD CHAR(80)%
,       02 ID_AND_NUMBER  CHAR(4)   %
,         03 ID           CHAR(3)   %
,        03 NUMBER        CHAR(1)   %
,      02 FILLER          CHAR (76)  %
;
     01 VOL_HEADER_RECORD     CHAR (80)
```

```
,      02 FILLER           CHAR(4)  %
,      02 VOL_ID           CHAR(6)  %
,      02 ACCESSIBILITY    CHAR(1)  %
,      02 RFS              CHAR(26) % SUPPOSED TO BE RESERVED BUT WE WILL
                                    %USE IT ANYWAY
,        03 MFID           CHAR(17) %"0" IF NO MULTIPLE FILE ID
                                    %"X0" FOR 17  IF SCRATCH
                                    %"BACKUP" FOR BACKUP
,        03 SYS_SYMBOL     CHAR(2)  % "17"
,        03 TAPE_TYPE           CHAR(1)    % 0= SCRATCH
                                           % 1= USER
                                           % 2= BACKUP
                                           % 3=LIBRARY
,        03 FILLER         CHAR(6)  %
,      02 OWNER_ID         CHAR(14) %
,      02 FILLER     CHAR(28)%
,      02 VERSION          CHAR(1)  % 1 FOR THIS STANDARD
;


RECORD
    01 HEADER_1_RECORD CHAR (80)
,      02  FILLER          CHAR(4)
,      02 FILE_ID          CHAR(17)
,      02 FILE_SET_ID      CHAR(6)
,      02 FILE_SECTION_NO  CHAR(4)
,      02 FILE_SEQ_NO      CHAR(4)
,      02 GENERATION_NO    CHAR(4)
,      02 GENERATION_VERSION.NO CHAR(2)
,      02 CREATION_DATE    CHAR(6)
,      02 EXPIRATION_DATE  CHAR(6)
,      02 ACCESSIBILITY    CHAR(1)
,      02 BLOCK_COUNT      CHAR(6) %HDR1="000000",ECV & EOF = REAL THING
,      02 SYSTEM_CODE      CHAR(13) %
,      02 FILLER           CHAR(7)  % RFS
;
                                    %HDR2,EOV2,EOF2
RECORD                              %
     01 HEADER2_RECORD       CHAR(80)   %
,      02 FILLER              CHAR(4)    %
,      02 RECORD_FORMAT       CHAR(1)    % F= FIXED,D=VARIABLE,S=SPANNED
,                                        % U= UNDEFINED
,      02 BLOCK_LENGTH        CHAR(5)    %
,      02 RECORD_LENGTH       CHAR(5)    %
,      02 RESV_SYSTEM_USE     CHAR(35)   %
,        03 DENSITY           CHAR(1)    % 0=800, 1=556,2=200,3=1600
,        03 SENTINAL          CHAR(1)    %
,        03 PARITY            CHAR(1)    % 0= ALPHA(EVEN),1=BINARY(ODD)
,        03 EXT_FORM          CHAR(1)    % 0= UNSPECIFIED
                                         % 1= BINARY
```

```
                                                % 2= ASCII
                                                % 3= BCL
                                                % 4= EBCDIC
,         03 FILLER             CHAR(31)      %
,       02 FILLER               CHAR(28)      % RFS
;
  %
  %
  %
```

COLD START VARIABLES

```
RECORD 1 SYSTEM_UNITS_ARRAY BIT(16),
    2 SYS_PCU              BIT(12),
      3 SYS_PC               BIT(7),
        4 SYS_P                BIT(3),
        4 SYS_C                BIT(4),
      3 FILLER               BIT(1),
      3 SYS_U                BIT(4),
    2 AVL_DISP             BIT(4);
DEFINE CSV_SIZE AS #1109#;
RECORD 1 COLD_START_VARIABLES BIT (CSV_SIZE),
    02 CLEAR_START_FLAGS                    BIT(40),
       03 CS_WHICH                             BIT(4),
       03 CS_TRACE                             BIT(4),
       03 CS_INTERP                            BIT(4),
       03 CS_MCP                               BIT(4),
       03 CS_GISMO                             BIT(4),
       03 CS_INIT                              BIT(4),
       03 CS_EMULATE                           BIT(4),
       03 CS_MICRO_MCP                         BIT(4),
    02 NAME_TABLE                           BIT(36),
    02 INTERP_DIC_ENTRIES                   BIT(24),
    02 CS_SIZE                              BIT(24),
    02 DUMP_FILE                            BIT(36),
    02 CSV_COLD_START_LEVEL                 BIT(24),
       03 L61_NAME_TABLE                       BIT(1),
       03 L10_0_NEW_SYS_DISK_TABLES            BIT(1),
       03 FILLER                               BIT(22),
    02 GISMO_TRACE_FLAGS                    BIT(24),
    02 DISK_CS_SIZE                         BIT(24),
    02 DUMP_FILE_SIZE                       BIT(16),
    02 CORRECTABLE_ERROR_TABLE_LENGTH       BIT(16),% 40 + 32*#ENTRIES
    02 MPF_TABLE                            BIT(36),
    02 LOG_MIX_INFO                         BIT(36),
    02 DISK_AVAIL                           BIT(36),
    02 DISK_DIRECTORY                       BIT(36),
    02 TEMP_TABLE                           BIT(36),
    02 SY_DAY                               BIT(5),
    02 SY_MONTH                             BIT(4),
    02 SY_YEAR                              BIT(7),
    02 SY_JDAY                              BIT(9),
    02 SY_TIME                              BIT(21),
       03 SY_HOUR                              BIT(5),
       03 SY_MIN                               BIT(6),
       03 SY_SEC                               BIT(6),
       03 SY_10THSEC                           BIT(4),
    02 SY_12HOUR                            BIT(5),
```

```
02 SY_DAYNAME                    CHAR(9),
02 SY_MERIDIAN                   CHAR(2),
02 SYSTEM_OPTIONS                BIT(80),
   03 LOG_OPTION                    BOOLEAN,
   03 CHARGE_OPTION                 BOOLEAN,
   03 LIB_OPTION                    BOOLEAN,
   03 OPEN_OPTION                   BOOLEAN,
   03 TERM_OPTION                   BOOLEAN,
   03 TIME_OPTION                   BOOLEAN,
   03 DATE_OPTION                   BOOLEAN,
   03 CLOSE_OPTION                  BOOLEAN,
   03 PBT_OPTION                    BOOLEAN,
   03 PBD_OPTION                    BOOLEAN,
   03 BOJ_OPTION                    BOOLEAN,
   03 EOJ_OPTION                    BOOLEAN,
   03 SCHM_OPTION                   BOOLEAN,
   03 LAB_OPTION                    BOOLEAN,
   03 RMOV_OPTION                   BOOLEAN,
   03 DUMP_OPTION                   BOOLEAN,
   03 ZIPP_OPTION                   BOOLEAN,
   03 MEM_OPTION                    BOOLEAN,
   03 SW01_OPTION                   BOOLEAN,
   03 SW02_OPTION                   BOOLEAN,
   03 SW03_OPTION                   BOOLEAN,
   03 TOUT_OPTION                   BOOLEAN,
   03 FILLER                        BOOLEAN, % UNUSED
   03 TRMD_OPTION                   BOOLEAN,
   03 DEBUG_OPTION                  BOOLEAN,
   03 DISP_OPTION                   BOOLEAN,
   03 SPOL_OPTION                   BOOLEAN,
   03 RMSG_OPTION                   BOOLEAN,
   03 SQRM_OPTION                   BOOLEAN,
   03 COPY_OPTION                   BOOLEAN,
   03 BREL_OPTION                   BOOLEAN,
   03 HPRI_OPTION                   BOOLEAN,
   03 THRASHING_OPTION              BOOLEAN,
   03 FLMP_OPTION                   BOOLEAN,
   03 VLCP_OPTION                   BOOLEAN,
   03 VLIO_OPTION                   BOOLEAN,
   03 BRGR_OPTION                   BOOLEAN,
02 FIRST_SCHED_ENTRY             BIT(36),
02 FIRST_WAITING_SCHED           BIT(36),
02 MIX_LIMIT                     BIT(8),%
02 SYSTEM_UNITS                  BIT(256),
   03 SUS (16)                     SYSTEM_UNITS_ARRAY,
02 SPOLOG_SIZE                   BIT(16),% 14 BITS ARE ENOUGH.,
02 NEXT_SPOLOG_REC               BIT(36),%NEXT SPOLOG SEGMENT,
02 SPOLOG_LAST_AREA              BIT(1),% SPO.LOG.FULL
02 SPOLOG_LAST_SECTORS           BIT(2),% NOT USED TILL 6.1
02 SPO_Q_CLEAR_START             BIT(48),% SPO.Q.POINTER.
```

```
    02 CRT_SPO_COLUMNS                  BIT(1),% SET UP AT COLD START TO
    02 SPO_DISPLAY_TIME                 BIT(1),%
    02 CRT_SPO_DIRN                     BIT(1),%
    02 SPO_SUPPRESS                     BIT(1);% IF ON THE DISPLAY ONLY I
 %
 %
DECLARE CSV COLD_START_VARIABLES REFERENCE;
 %
 %
```

DISK COLD START VARIABLES

```
 %  THE FOLLOWING REMAPS PRO_ELEMENT IN DISK_COLDSTART_VARIABLES
 %
RECORD 1 PRO_ARRAY              BIT(16),%
    2 PCU                       BIT(12),%
    2 LABEL_TYPE                BIT(2),%
          % 0 = ANSII
          % 1 = UNLABELED
          % 2 = BURROUGHS
    2 TRANSLATE                 BIT(1),% 0=EBCDIC   1=ASCII
    2 CLEAR_START               BIT(1);%
 %
DEFINE DCSV_SIZE AS #1207#;
 %
RECORD  01 DISK_CS_VARIABLES BIT(DCSV_SIZE),
    02 MASTER_IOAT                      DSK_ADR,
    02 MASTER_DISK_AVAIL                DSK_ADR,
    02 NEXT_LOG_REC                     DSK_ADR,
    02 LG_SIZE                          WORD,
    02 NEXT_ELOG                        DSK_ADR,
    02 ELOG_SIZE                        WORD,
    02 JOB_NO                           WORD,
    02 PBD_NO                           WORD,
    02 SPO_Q_SIZE                       WORD,
    02 CTLDCK_NO                        WORD,
    02 LOG_NO                           WORD,
    02 Q_DISK                           DSK_ADR,
    02 TRACE_FPB                        DSK_ADR,
    02 AUTO_MASK_ARRAY                  BIT(28),% 4 OF 7 EACH=PRT PC
       03 AUTO_MASK (4)                   BIT(7),
    02 AB_NUMBER                        BIT(3),% NUM OF SYSTEM/BACKUPS
    02 FILLER                           BIT(5),%
    02 PBD_BLCKS_AREA                   WORD,
    02 LG_LAST_AREA                     BOOLEAN,
    02 ELOG_LAST_AREA                   BOOLEAN,
    02 FILLER                           BIT (2),
    02 PBD_DESIGNATION                  CHAR(10),
    02 SPO_Q                            DSK_ADR,
    02 PROTECTED_UNITS                  BIT(256), %
       03 PRO_ELEMENT (16)                PRO_ARRAY,
    02 SYS_LOG_NUMBER                   BIT(24),
    02 JOB_ACCTING_NUMBER               BIT(24),
    02 SESSION_NR                       BIT (16),
    02 XM_TABLE_DISK_ADDRESS            DSK_ADR,
    02 DCSV_NSEC_DISABL_THRASH_FAULT    BOOLEAN,
    02 DCSV_OVERLAY_RATE                BIT(6),
```

```
    02 DCSV_THRASHING_SENSITIVITY       BIT(8),
    02 DCSV_HOSTNAME                    CHAR (17),% BNA.
    02 NEXT_HOSTNAME                    CHAR (17);
  %
DECLARE DCSV DISK_CS_VARIABLES REFERENCE;
  %
  %
```

MEMORY LINK

```
DEFINE  MEMORY_LINK_SIZE AS #187#;
RECORD 1 MEMORY_LINK  BIT(MEMORY_LINK_SIZE),
           2 ML_DISK                    DSK_ADR,
           2 ML_GROUP                   BIT(47),
             3 ML_POINTER                 ADDRESS,
             3 ML_JOB_NUMBER              BIT(16),
             3 ML_TYPE                    BIT(6),
             3 ML_SAVE                    BIT(1),
           2 ML_SIZE                    BIT(24),
           2 ML_PRIORITY_FIELD          BIT(30),
             3 ML_DK_INTERVAL             BIT(10),
             3 ML_CURRENT_DK_INT          BIT(10),
             3 ML_INCOMING_PRIORITY       BIT(5),
             3 ML_RESIDENCE_PRIORITY      BIT(5),
               4 ML_RP_WHOLE                BIT(4),
               4 ML_RP_FRACTION             BIT(1),
           2 ML_FRONT                   BIT(24),
           2 ML_BACK                    BIT(24),
           2 ML_USAGE_BITS              BIT(2),
             3 ML_PREVIOUS_SCAN_TOUCH     BIT(1),
             3 ML_CURRENT_SCAN_TOUCH      BIT(1);


RECORD
   01 Q_MEMORY_LINK           BIT(235) %MEMORY_LINK_SIZE+48
,     02 MEM_LINK             BIT(MEMORY_LINK_SIZE)
,     02 Q_ML_F_AVL           ADDRESS
,     02 Q_ML_B_AVL           ADDRESS
;
 %
```

ML TYPE

```
DEFINE        % TYPES FOR "ML.TYPE"
 %
 %   CAUTION:  WHEN ADDING OR CHANGING TYPES BE SURE TO UPDATE THE TWO
 %             DEFINES - "MUST_IT_GO_ABOVE_FENCE" & "ALLOCATE_FROM_LO".
 %
    CODE                              AS #0#%
,   DATA                              AS #1#%
,   AVAILABLE                         AS #2#%
,   RN_S                              AS #3#%
,   MCP_TEMP                          AS #4#%
,   USER_FILE                         AS #5#%
,   SEG_DICTV                         AS #6#%
,   MICROCODE                         AS #7#%
,   DICT_MASTER                       AS #8#%
,   QUEUE_DIRECTORY_TYPE              AS #9#%
,   MSG_BUFFERV                       AS #10#%
,   MESSAGE_LIST_TYPE                 AS #11#%
,   TO_BE_FORGOTTEN                   AS #12#%
,   DATA_SEG                          AS #13#%
,   DMS_BUFFER                        AS #14#%
,   TERMINATING_LINK                  AS #15#%
,   MCP_PERM                          AS #16#%
,   PSR_MEM                           AS #17#%
,   MCP_IOAT                          AS #18#%
,   DISK_HEADER                       AS #19#%
,   PACK_MEM                          AS #20#%
,   SD_CNTNR                          AS #21#%
,   SCHED_MEM                         AS #22#%
,   SORT_MEM                          AS #23#%
,   DCH_MEM                           AS #24#%
,   MICROCODE_NON_OVERLAYABLE         AS #25#%
,   QUEUE_AVL_BUF_V                   AS #26#%
,   DMS_DISK_HDR                      AS #27#%
,   DMS_STRUCTURE                     AS #28#%
,   DMS_TEMP                          AS #29#%
,   DMS_GLOBALS                       AS #30#%
,   DMS_TEMP_LOCK_DESCR               AS #31#%
,   XM_MEMORY                         AS #32#%
,   PERM_SPO_BUFF                     AS #33#%
,   DMS_WORKAREA                      AS #34#%
,   I_S_CURRENT                       AS #35#%
,   INTERP_DATA                       AS #36#%
,   I_S_BUFFER                        AS #37#%
,   I_S_STRUCTURE                     AS #38#%
,   RUN_UNIT                          AS #39#%
```

```
;%
 %
 %
 %
 %
 %
 %
```

SYSTEM DESCRIPTORS

```
DECLARE
    01 SYSTEM_DESCRIPTOR TEMPLATE BIT(SY_SIZE);
  %
  %
  %
  %
RECORD
    01 SYSTEM_DESCR          BIT(SY_SIZE),
       02 SY_IN_USE          BIT(1),    % TO HELP MEMORY MANAGEMENT
       02 SY_MEDIA           BIT(1),    % 0=DISK, 1=S-MEMORY
       02 SY_LOCK            BIT(1),    %
       02 SY_IN_PROCESS      BIT(1),    % TRUE IF THERE IS AN I/O IN
                                        % PROCESS FOR THE INFORMATION
                                        % REPRESENTED BY THIS DESCRIPTOR.
                                        % IF TRUE, "SY.CORE" CONTAINS A
                                        % POINTER TO THE I/O DESCRIPTOR.
       02 SY_INITIAL          BIT(1),    % "ADDRESS" IS READ-ONLY MOTHER
                                        % COPY, HENCE IF "WRITE" THEN GET
                                        % NEW DISK AND REPLACE ADDRESS.
       02 SY_FILE             BIT(1),    % THE OBJECT OF THIS DESCRIPTOR
                                        % IS A FILE WHOSE USERCOUNT MUST
                                        % BE DECREMENTED WHEN THIS
                                        % DESCRIPTOR IS RETIRED.
       02 SY_DK_FACTOR        BIT(3),    % MEMORY DECAY FACTOR
       02 SY_SEG_PG           BIT(7),    % MCP MEMORY ACTIVITY AUDITING
       02 SY_TYPE             BIT(4),    % UNITS FOR SY.LENGTH.
                                        % 0 = BITS
                                        % 1 = DIGITS (4 BIT)
                                        % 2 = CHARACTERS (8 BIT)
                                        % 3 = NORMAL DESCRIPTORS
                                        % 4 = DISK SEGMENTS
                                        % 5 = SYSTEM DESCRIPTORS
                                        % 6 = SYSTEM INTRINSIC
                                        % 7 = INDIRECT REFERENCE
                                        %     ADDRESS GIVES RELATIVE
                                        %     DISPLACEMENT IN BITS
                                        %     (SIGNED NUMBER).
                                        % 8 = MICROS
       02 SY_ADDRESS          BIT(36),   %
          03 FILLER           BIT(12),   % PORT,CHANNEL AND UNIT.
          03 SY_CORE           BIT(24),  % CORE, OR ADDRESS WITHIN UNIT.
       02 SY_LENGTH           BIT(24);   % NUMBER OF UNITS, AS DETERMINED
                                        % BY SY.TYPE.
  %
  %
  %
```

```
    DEFINE ND.DECLARATION AS#
    DECLARE
    01 DUMMY REMAPS NORMAL.DESCRIPTOR BIT(ND.SIZE),
       02 ND.DK.FACTOR        BIT(3),
       02 FILLER              BIT(6),
       02 ND.CORE             BIT(24),
       02 ND.TYPE             BIT(3),
       02 ND.LENGTH           BIT(24);#;
%
%
%
```

RUN STRUCTURE STATUS TYPES

```
 %
 %
 %
 %   ************************************************************************
 %   *                                                                      *
 %   *                  I M P O R T A N T                                   *
 %   *                  - - - - - - - - -                                   *
 %   *                                                                      *
 %   *      THE CAUSE ROUTINE IS DRIVEN BY "RS_STATUS" AND                  *
 %   *      MUST BE UPDATED IF THIS LIST IS CHANGED                         *
 %   *      LIKEWISE, "MAX_REASON" (A DEFINE) MUST EQUAL THE                *
 %   *      LARGEST NUMBER IN THE LIST.                                     *
 %   *                                                                      *
 %   ************************************************************************
 %
 %
DEFINE MAX_REASON AS #61#;
 %
 %
DEFINE   %    "RS_STATUS" TYPES
    EXECUTING                   AS #0#
,   NO_FILE                     AS #1#
,   NO_USER_DISK                AS #2#
,   DUPLICATE_LIBRARY           AS #3#
,   DUPLICATE_INPUT_FILE        AS #4#
,   POSSIBLE_DUP                AS #5#
,   WAITING_FOR_HARDWARE        AS #6#
,   PROGRAM_STOPPED             AS #7#
,   WAITING_IO_COMPLETE         AS #8#
,   WTG_DATACOMM_MSG            AS #9#%
,   WAITING_OVERLAY             AS #10#
,   WAITING_KBD_IN              AS #11#
,   HDWR_NOT_READY              AS #12#
,   WAITING_OPERATOR_ACTION     AS #13#
,   WAITING_CLOSE               AS #14#
,   WAITING_DS_OR_DP            AS #15#
,   NO_MPF_PACK                 AS #16#
,   NO_FILE_ON_DISK             AS #17#%
,   WAITING_FOR_LOCKED_FILE     AS #18#
,   WAITING_Q_IS_FULL           AS #19#
,   WAIT_STATUS                 AS #20#
,   NOMEM_WAITING_COMM_Q        AS #21#
,   NOMEM_WAITING_READY_Q       AS #22#
,   WAITING_PORT_OPEN           AS #23#
,   WTG_PGM_CALL                AS #24#
,   WAITING_TIME_COMM_Q         AS #25#
```

```
,    WAITING_TIME_READY_Q       AS #26#
,    WAITING_RECEIVE            AS #27#
,    WTG_DATACOMM_OPN           AS #28#%
,    TERMINATING                AS #29#
,    IN_READY_Q                 AS #30#
,    IN_COMM_Q                  AS #31#
,    STOPPED_FOR_SORT           AS #32#
,    WTG_DC_DSK_CMPLT           AS #33#%
,    WTG_DATACOMM_DSK           AS #34#%
,    NO_CONTROLLER              AS #35#%
,    NO_OUTPUT_PACK             AS #36#
,    VSORT_QSORT_NOT_PRESENT    AS #37#
,    NO_SORT_INPUT_FILE         AS #38#
,    WAITING_CONTENTION         AS #39#
,    WAITING_SYNCPOINT          AS #40#
,    WAITING_RECOVERY           AS #41#
,    WAITING_NEW_AUDIT          AS #42#
,    WAITING_SORTER_IO          AS #43#
,    TERMINATING_WAITING_IO     AS #44#
,    CLOSING_WAITING_IO         AS #45#
,    WAITING_FORMS              AS #46#
,    NO_TRANSLATE_FILE          AS #47#
,    MF_SEARCHING               AS #48#  %
,    NO_DMS_FILE                AS #49#
,    NO_DMS_DICTIONARY          AS #50#
,    WTG_DMS_REORGANIZATION     AS #51#
,    WTG_INACTIVE_DATA_BASE     AS #52#
,    NO_USERCODE                AS #53#
,    WAITING_TO_BE_CALLED       AS #54#
,    WTG_PROGRAM_EXIT           AS #55#
,    WTG_CALLED_PGM_BOJ         AS #56#
,    WTG_REL_AREA_INIT          AS #57#
,    WTG_DATACOMM_RESULT        AS #58#
,    WTG_BEGINNING_LABEL        AS #59#%
,    NO_PROGRAM                 AS #60#%
,    NO_HOST_SERVICES           AS #61#
;
 %
```

RUN STRUCTURE NUCLEUS

```
DEFINE RS_N_SIZE AS #2384#;
 %
RECORD
 %
1 RS_NUCLEUS                          BIT(RS_N_SIZE)
 %
 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 % THE FOLLOWING ARE USED BY THE INTERPRETERS, MMCP, GISMO AND THE SMCP
 %
, 2 RS_COMMUNICATE_MSG_PTR            BIT(48)
              %CONTAINS EITHER AN SDL DESCRIPTOR THAT POINTS TO A
              %COMMUNICATE MESSAGE OR THE MESSAGE ITSELF
,   3 RS_ITYPE                        BIT(2)
              %DEFINES THE USE OF COMMUNICATE.MSG.PTR
              %     00 = PROGRAM INTERNAL INTERRUPT
              %     01 = COMMUNICATE
              %     10 = UNDEFINED
              %     11 = TERMINATING
              %SEE PROCEDURE IH FOR FURTHER CLARIFICATION
,   3 RS_INMBR                        BIT(6)
              %INTERRUPT NUMBER IF RS.ITYPE=00
              %SEE IH FOR DEFINITION OF VALUES
,   3 RS_ILENGTH                      BIT(16)
              %LENGTH OF COMMUNICATE MESSAGE IF RS.ITYPE=01
,   3 RS_IADDRESS                     BIT(24)
              %ADDRESS OF COMMUNICATE MESSAGE IF RS.ITYPE=01
              %NOT USED IF RS.ITYPE=10 OR 11
              %SEE IH FOR DEFINITION WHEN RS.ITYPE=00
, 2 RS_COMMUNICATE_LR                 BIT(24)
              %LIMIT REGISTER OF THE RUN STRUCTURE THAT THE
              %COMMUNICATE IS DIRECTED TO
, 2 RS_REINSTATE_MSG_PTR              BIT(48)
              %SELF-RELATIVE SDL TYPE DESCRIPTOR USED TO PASS THE RESULT
              %OF A COMMUNICATE FROM THE MCP TO A NORMAL STATE PROGRAM.
              %SEE EACH COMMUNICATE FOR DEFINITION OF VALUES.
, 2 RS_MY_BASE                        BIT(24)
              %BASE REGISTER OF THIS RUN STRUCTURE
, 2 RS_MY_LIMIT                       BIT(24)
              %LIMIT REGISTER OF THIS RUN STRUCTURE
, 2 RS_MCP_BIT                        BIT(1)
              %INDICATES TO THE INTERPRETER THAT THIS RUN STRUCTURE
              %BELONGS TO A CONTROL STATE JOB
, 2 RS_NIP                            BIT(32)
              %PAGE, SEGMENT AND DISPLACEMENT OF THE NEXT EXECUTABLE
              %INSTRUCTION FOR THIS JOB. SEG(6), P(6), D(20).
, 2 RS_SEG_DIC_PTR                    BIT(24)
```

```
                %ADDRESS OF THE MASTER CODE SEGMENT DICTIONARY
* 2 RS_DATA_DIC                 BIT(24)
                %ADDRESS OF THE DATA SEGMENT DICTIONARY
* 2 RS_FIB_DIC                  BIT(24)
                %MEMORY ADDRESS OF THE FIB DICTIONARY
* 2 RS_PAGED_ARRAY_OVERLAY      BIT(6)
                %SEGMENT NUMBER OF THE SDL PAGED ARRAY HANDLER OVERLAY
                %IF REQUIRED FOR THIS JOB. ALWAYS PAGE 0.
                %FORMERLY RS.INTRINSICS.LOC
* 2 RS_SPAD_SIZE                BIT(16)
                %SIZE IN BITS OF SCRATCH PAD FOR THE M-MACHINE.
                %FOR B1700/B1800 IT WILL BE 768
* 2 RS_SPAD_PTR                 BIT(24)
                %ADDRESS OF SCRATCH PAD IN S-MEMORY
* 2 RS_INTERP_DATA_SIZE         BIT(24)
                %LENGTH IN BITS OF INTERPRETER DATA SPACE
* 2 RS_INTERP_DATA_ADDR         BIT(24)
                %ABSOLUTE ADDRESS OF INTERPRETER DATA SPACE
* 2 RS_SPO_INPUT_PRESENT        BIT(1)
                %INDICATES THAT AN AX WAS DONE
                %FORMERLY RS.EXTERNAL.INTERRUPT.BIT
* 2 RS_TRACE_BUF_ADDR           BIT(24)
                %MEMORY ADDRESS OF THE TRACE BUFFER IF THIS NORMAL STATE
                %JOB IS TRACING.
* 2 RS_TRACE_BITS               BIT(8)
                %FLAGS INDICATING WHAT TYPE OF TRACE IS TO BE PERFORMED
* 2 RS_SWITCHES                 BIT(40)
                %10 4-BIT SWITCHES.    SW0-9
* 2 RS_TIME                     BIT(20)
                %TOTAL ELAPSED PROCESSOR TIME FOR THIS JOB
* 2 RS_TIME_ACCUM               BIT(24)
                %TOTAL ACCUMULATED PROCESSOR TIME (500 NANO TICKS)
* 2 RS_IPC_DICT                 BIT(24)
                %ABSOLUTE ADDRESS OF THE IPC.DICTIONARY FOLLOWING THIS
                %RS.NUCLEUS. (FOR IPC)
* 2 RS_IPC_DICT_SIZE            BIT(16)
                %NUMBER OF ENTRIES IN THE IPC.DICTIONARY
* 2 RS_CALLERS_LR               BIT(24)
                %LIMIT REGISTER OF THIS JOBS CALLER
* 2 RS_LAST_LIO_STATUS_SIZE     BIT(16)
                %SIZE OF LAST.LIO.STATUS MASK
* 2 RS_LAST_LIO_STATUS_PTR      BIT(24)
                %ADDRESS OF LAST.LIO.STATUS MASK
* 2 RS_Q_LINK                   BIT(24)
                %POINTER TO THE NEXT JOBS RS.NUCLEUS
                %FIRST.QUEUE POINTS TO 1ST JOB. LAST JOB CONTAINS @FFFFFF@
* 2 RS_LAST_TOP                 BIT(1)
                %
* 2 RS_DATA_DIC_SIZE            BIT(23)
                %NUMBER OF DATA DICTIONARY ENTRIES.
```

```
,  2 RS_Q_IDENT                    BIT(24)
                %THE QUEUE THAT THIS JOB IS CURRENTLY IN
                %     0 = READY.Q
                %     1 = S.COMM.Q
                %     3 = EXTERMINATE.Q
                %    10 = M.COMM.Q
                %    11 = WATE.Q
                %    -2 = NOT QUEUED
,  2 RS_NEXT_Q                     BIT(24)
                %IF THIS JOB IS IN THE WATE.Q, THE QUEUE IT SHOULD BE
                %PLACED IN WHEN IT IS CAUSED.
,  2 RS_STATUS                     BIT(24)
                %GIVES THE CURRENT STATUS OF THE JOB.
                %REF STATUS DEFINES AT LINE 02722000
,  2 RS_PRIORITY_INTEGER           BIT(4)
                %PROCESSOR PRIORITY - 0-15 ALLOWED
,  2 RS_INTERP_ID                  BIT(5)
                %INTERPRETER NUMBER FOR THIS JOB. INDEX INTO THE
                %INTERPRETER DICTIONARY.
,  2 RS_MEDIA                      BIT(1)
                % 0 = JOB ROLLED OUT TO DISK
                % 1 = JOB IN S-MEMORY
,  2 RS_JOB_NUMBER_IN_DECIMAL      BIT(16)
                % E.G. JOB NUMBER 1753 WOULD BE @17530
,  2 RS_PAUSE                      BIT(24)
                %TIME TO WAKE THIS JOB IF SLEEPING
,  2 RS_WAIT_LEN                   BIT(12)
                %LENGTH OF RS.EVENT.SPACE
,  2 RS_WAIT_LOC                   BIT(24)
                %ADDRESS OF RS.EVENT.SPACE
,  2 RS_DISABLE_INTERRUPTS         BIT(6)
                %IF THIS FIELD IS GREATER THAN 0 THEN THIS JOB MAY NOT
                %BE INTERRUPTED BY HIGH PRIORITY INTERRUPTS.
,  2 RS_USE_FLAG                   BIT(1)
                %IF TRUE, JOB IS CURRENTLY ACTIVE IN A
                %USE ROUTINE
,  2 RS_REPORT_EV_INX              BIT(1)
                %USED BY PROCESSES THAT WISH TO HANG JOBS AND HAVE THE
                %EVENT WHICH WAKES UP THE JOB REPORTED IN THE RS.
                %(USED BY M.WAIT AND COMPLEX.WAIT).
,  2 RS_STATE_LIGHT                BIT(16)
                %USED BY THE LAMP CODE IN GISMO TO DISPLAY ACTIVITIES
                %BY JOB.
,     3 RS_VARIABLE_LAMP_CPU       BIT(2)
                %USED TO DISPLAY JOB CPU ACTIVITY
,     3 RS_VARIABLE_LAMP_CODE_OVLY BIT(2)
                %USED TO DISPLAY JOB CODE OVERLAYS
,     3 RS_VARIABLE_LAMP_DATA_OVLY BIT(2)
                %USED TO DISPLAY JOB DATA OVERLAYS
,     3 FILLER                     BIT(10)
```

```
                %
* 2 RS_SLAVE_BLOCKED_COUNT          BIT(6)
                %NUMBER OF BLOCKS ON THE SLAVE SCHEDULER. (DUAL CPU).
* 2 RS_JOB_NUMBER                   BIT(16)
                %CONTAINS THE JOB NUMBER ASSIGNED TO THIS JOB. ASSIGNED
                %WHEN THE JOB IS SCHEDULED. JOB NUMBER IS USED ON ANY
                %SPO INPUT MESSAGE THAT REQUIRES JOB INDENTIFICATION.
                %BEGINS WITH 1 AND WRAPS AROUND AT 9999.
* 2 RS_ABORT                        BIT(2)
                % 0 = RUNNING
                % 1 = DS OR DP-ED
                % 2 = CANCELED
                % 3 = DUE TO DEATH IN FAMILY
* 2 RS_DC_IO_COMPLETE               BIT(1)
                %THIS EVENT IS CAUSED WHENEVER A DATA COMM I/O OR
                %AN INITIALIZER I/O COMES COMPLETE
* 2 RS_DATA_COMM                    BIT(1)
                %IF TRUE, JOB HAS DONE A DC.INITIATE.IO
* 2 RS_SORTER_FLOWING               BIT(1)
                %MICR JOB WITH READER/SORTER CURRENTLY IN FLOW MODE
* 2 RS_TO_BE_ROLLED_OUT             BIT(1)
                %IF TRUE, JOB IS A CANDIDATE FOR ROLLOUT - DO IT NEXT
                %N.SECOND
* 2 RS_NOT_A_ROLLOUT_CANDIDATE      BIT(1)
                %IF TRUE, JOB HAS BEEN HUNG BUT CANNOT BE ROLLED OUT.
* 2 RS_INTERVENTION                 BIT(1)
                %SMCP NEEDS TO DO SOMETHING TO THIS JOB BEFORE
                %THE MMCP CAN HAVE IT. (USUALLY ROLLIN).
* 2 RS_M_PROBLEM                    BIT(48)
                %REASON WHY THE MMCP WOULD TURN CONTROL OF THIS
                %JOB OVER TO THE SMCP
*   3 RS_M_PROBLEM_TYPE             FIXED
                % 1=LIO PROBLEM (SEE PARAMETERS)
                % 3=FIB DICT NOT PRESENT
                % 5=RS.INTERVENTION SET
                % 7=DUMP COMMUNICATE SENT TO MMCP
                %20=MMCP PAGE FAULT (PARAMETER=SEG DESC ADDR)
                %30=INVALID COMPLEX.WAIT COMMUNICATE
                %31=NO SPO.QUEUE
*   3 RS_M_PROBLEM_PARAMETER        BIT(24)
                %   **** TYPE 1 ****
                % 1=IRRECOVERABLE EXCEPTION
                % 2=FIB NOT OPEN
                % 3=WRONG POSITION
                % 4=NEED NEW AREA
                % 5=INVALID CHARACTER ON PSEUDO READER FILE
                % 6=EOF
                % 7=AREA OUT OF BOUNDS
                % 8=DISK FILE HEADER INDICATES A MULTI PACK FILE
                % 9=AREA NOT PRESENT
```

```
                %10=LOGICAL I/O ALLOWED ONLY FROM SMCP
                %11=DISK FILE HEADER NOT PRESENT
                %12=INVALID FILE ACCESS
                %13=VARIABLE RECD SIZE BELOW BOUNDS
                %14=VARIABLE RECD SIZE ABOVE BOUNDS
                %15=VARIABLE RECD SIZE INVALID ON INPUT
                %16=USER DATA OUTSIDE BASE-LIMIT
                %17=EMULATOR TAPE IRRECOVERABLE EXCEPTION
                %18=EMULATOR TAPE ILLEGAL INITIATE
                %19=EMULATOR TAPE ILLEGAL FETCH
                %20=EMULATOR TAPE OVERLAP
                %21=EMULATOR TAPE ILLEGAL OPCODE
                %22=EMULATOR TAPE ILLEGAL ERROR MASK
                %23=EMULATOR TAPE ILLEGAL ACCESS
                %  .
                %  .
                %  .
* 2 RS_SPO_Q_KEY                    BIT(24)
                %POINTS AT THE QUEUE DESCRIPTOR DESCRIBING THE USERS
                %ACCEPT QUEUE.
* 2 RS_FILE                         BIT(8)
                %IF THE JOB IS HUNG FOR ANY PROBLEM WITH A FILE, THIS
                %CONTAINS THE INDEX INTO THE FIB DICTIONARY FOR THE
                %FILE IN QUESTION.
* 2 RS_RUN_UNIT                     BIT(16)
                %JOB NUMBER OF THE PARENT OF THIS RUN UNIT (FOR IPC)
* 2 RS_RUN_UNIT_LINK                BIT(16)
                %JOB NUMBER OF THIS JOBS CALLER (FOR IPC)
* 2 RS_IPC_PARAMETER_LIST           BIT(24)
                %ABSOLUTE ADDRESS OF THE IPC.PARAMETER.LIST
* 2 RS_EXECUTE_TYPE                 BIT(4)
                %  1 = EXECUTE
                %  2 = COMPILE AND GO
                %  3 = COMPILE FOR SYNTAX
                %  4 = COMPILE TO LIBRARY
                %  5 = COMPILE AND SAVE
                %  6 = GO PART OF COMPILE AND GO
                %  7 = GO PART OF COMPILE AND SAVE
* 2 RS_NAME                         CHARACTER(30)
                %NAME OF THIS JOB
* 2 RS_IPC_EVENT                    BIT(1)
                %DUMMY EVENT FOR ANY HANG FOR IPC
* 2 RS_CANCELED                     BIT(1)
                %A CANCEL COMMUNICATE HAS BEEN ISSUED AGAINST THIS JOB
* 2 RS_EVENT_SPACE                  BIT(24*15)
                %REPRESENTS THE LIST OF EVENTS ON WHICH A JOB IN THE
                %WAIT.Q IS WAITING.
* 2 RS_BOJ_TO_EOJ_FREEZE            BIT(1)
                %IF TRUE, JOB WAS EXECUTED WITH FREEZE AND CAN NEVER BE
                %ROLLED OUT
```

```
, 2 RS_TEMPORARY_FREEZE                 BIT(8)
              %COUNTER THAT IS BUMPED EACH TIME A FREEZE IS DONE AND
              %DECREMENTED FOR EACH UNFREEZE. CHANGED BY FREEZE
              %COMMUNICATE, REMOTE FILE OPEN, INITIALIZER I/O, MICR
              %OPEN AND CLOSE.
, 2 RS_PROG_PTR                         BIT(36)
              %DISK ADDRESS OF THIS CODE FILE
, 2 RS_LOG_PTR                          BIT(36)
              %DISK ADDRESS OF WORKING PPB AND FPB-S
, 2 RS_DISK                             BIT(36)
              %DISK ADDRESSS OF JOB WHEN ROLLED OUT
, 2 RS_JOB_ACCTING_NO                   BIT(24)
              %A UNIQUE ID NUMBER FOR EACH JOB. RESET ONLY BY
              %COLDSTART. INCREMENTED BY 1 EACH TIME A JOB ENTERS THE
              %SCHEDULE. USED BY TABS.
, 2 RS_NUMBER_FILES                     BIT(8)
              %MAXIMUM NUMBER OF FPB-S DECLARED BY THIS PROGRAM.
, 2 RS_TYPE                             BIT(6)
              %HARDWARE TYPE REQUIRED TO RESOLVE MISSING HARDWARE
, 2 RS_TRACE_FIB                        BIT(8)
              %FILE NUMBER USED FOR TRACE. INDEX INTO THE
              %FIB.DICTIONARY
, 2 RS_SER_NU                           BIT(24)
              %SERIAL NUMBER OF A DISK PACK IF THIS JOB IS WAITING
              %FOR A BASE OR CONTINUATION PACK FOR MULTI PACK FILES
, 2 RS_UNIT_INDEX                       BIT(24)
              %ADDRESS OF IOAT OF DEVICE INDICATED BY IL,OU,FM,UL
, 2 RS_MCP_USE                          BIT(1)
              %IF TRUE, MCP IS WAITING FOR AN EVENT FLAGGED BY
              %RS.BOOLEANS TO OCCUR.
, 2 RS_BOOLEANS                         BIT(24)
              %USED BY THE SMCP TO INDICATE ACTIONS AVAILABLE TO
              %SOLVE OPEN AND CLOSE PROBLEMS
,   3 RS_IL                             BIT(1)
,   3 RS_UL                             BIT(1)
,   3 RS_OF                             BIT(1)
,   3 RS_FR                             BIT(1)
,   3 RS_FM                             BIT(1)
,   3 RS_OU                             BIT(1)
,   3 RS_OK                             BIT(1)
,   3 RS_RM                             BIT(1)
,   3 RS_MR                             BIT(1)
,   3 FILLER                            BIT(15)
, 2 RS_MEMORY_PRIORITY                  BIT(5)
              %MEMORY PRIORITY * 2  - 0-15 ALLOWED
, 2 RS_SWEEPS_BEFORE_DECAY              BIT(10)
              %NUMBER OF MEM.SWEEP.INTERVALS BEFORE IMPORTANT CODE
              %SEGMENTS WILL DECAY.
, 2 RS_FORCED_SUSPENSION                BIT(1)
              %IF TRUE, JOB HAS BEEN STOPPED BY AN "ST EOJ" MESSAGE
```

```
, 2 RS_LENGTH                       BIT(24)
              %LENGTH IN BITS OF THIS RUN STRUCTURE.
              %INCLUDES BASE-LIMIT, RS.NUCLEUS, FIB.DICT, DATA.DICT,
              %SCRATCH.PAD, PATH.DICT, IPC.PARAMETER.TABLE,
              %OVERLAY.DESCRIPTOR
, 2 RS_CODE_OVLY_COUNT              BIT(20)
              %NUMBER OF DISK ACCESSES FOR CODE OVERLAYS
, 2 RS_DATA_OVLY_COUNT              BIT(20)
              %NUMBER OF DISK ACCESSES FOR DATA OVERLAYS
, 2 RS_LINKS                        BIT(1)
              %IF TRUE, DYNAMIC SPACE CONTAINS MEMORY LINKS.
, 2 RS_LAST_OVLY                    BIT(24)
              %LEFT OFF POINTER FOR MEMORY MGMT. USED ONLY ON JOBS
              %WITH DATA OVERLAYS
, 2 RS_DATA_OVERLAYS                BIT(24)
              %ADDRESS OF DYNAMIC SPACE WITHIN BASE-LIMIT
              %FIRST LINK FOR MEMORY MGMT OF DATA OVERLAYS
, 2 RS_LAST_LINK                    BIT(24)
              %ADDRESS OF LAST MEMORY LINK WITHIN DYNAMIC SPACE.
              %USED FOR MEMORY MGMT
, 2 RS_OVLY_DISK_BASE               BIT(36)
              %DISK ADDRESS OF BEGINNING OF DATA OVERLAY SPACE
, 2 RS_OVLY_DISK_PTR                BIT(24)
              %INDEX INTO THE DATA OVERLAY AREA ON DISK
, 2 RS_OVLY_DISK_SIZE               BIT(24)
              %NUMBER OF DISK SEGMENTS RESERVED FOR DATA OVERLAYS
, 2 RS_PREVENT_MOVE                 BIT(1)
              %RS.NUCLEUS CANNOT BE MOVED WHEN JOB IS ROLLED OUT
, 2 RS_DISPLACED                    FIXED
              %DISTANCE REMNANT HAS BEEN MOVED WHEN THIS JOB
              %WAS ROLLED OUT.
, 2 RS_ROLLIN_IN_PROCESS            BIT(1)
              %IF TRUE, THE SMCP IS ROLLING THIS JOB INTO MEMORY
, 2 RS_ROLLOUT_COMPLETE             BIT(1)
              %THIS JOB WAS ROLLED OUT AT LEAST ONCE.
, 2 RS_ROLLOUT_IN_PROCESS           BIT(1)
              %THE SMCP IS CURRENTLY ROLLING THIS JOB OUT
, 2 RS_PROTECTED                    BIT(1)
              %IF TRUE, JOB IS LOCKED -     NEED LP- TO DS
, 2 RS_TO_BE_STOPPED                BIT(1)
              %IF TRUE, AN ST WAS ISSUED ON THIS JOB. IT IS TO BE
              %STOPPED WHEN CONVENIENT.
, 2 RS_STOPPED                      BIT(1)
              %IF TRUE, JOB HAS BEEN STOPPED BY ST
, 2 RS_SIZECHANGE                   BIT(1)
              %IF TRUE, THIS JOBS SCRATCH PAD IS BEING CHANGED.
, 2 RS_SD_PTR_FLAG                  BIT(1)
              % 0 = RS.SEG.DIC.PTR CONTAINS ADDRESS OF DICTIONARY
              %     CONTAINER.
              % 1 = RS.SEG.DIC.PTR CONTAINS ADDRESS OF SEGMENT
```

```
                %     DICTIONARY ITSELF.
* 2 RS_DONT_REENTER                BIT(1)
                %IF TRUE, THIS JOB CANNOT SHARE ITS SEGMENT DICTIONARY
* 2 RS_PAGED_DICT                  BIT(1)
                %INDICATES THAT CODE SEGMENT DICTIONARY IS PAGED
* 2 RS_EMULATOR_BITS               BIT(4)
                %USED BY THE B1700 EMULATOR
* 2 RS_EMULATOR_TAPE               BIT(8)
                %NUMBER OF EMULATOR TAPE FILES CURRENTLY OPEN
* 2 RS_PRIVILIGED                  BIT(1)
                %IF TRUE, JOB HAS A PRIVILIGED USERCODE
* 2 RS_APPARITION                  BIT(1)
                %IF TRUE, THIS JOB HAS CALLED ANOTHER JOB VIA SORT OR
                %PGM.CALLER AND IS WAITING FOR ITS COMPLETION.
* 2 RS_PARENT_QUEUE                BIT(24)
                %QUEUE ADDRESS OF THE PARENT PROGRAM ASSIGNED TO THIS
                %JOB. CERTAIN MESSAGES WILL BE PUT INTO THIS QUEUE.
* 2 RS_LOG_SPO                     BIT(1)
                %IF TRUE, SPO MESSAGES GENERATED BY THIS JOB ARE TO
                %BE PUT INTO PARENT QUEUE.
* 2 RS_USERCODE                    BIT(10)
                %INDEX INTO THE USERCODE TABLE FOR THE USERCODE THAT
                %THIS JOB IS RUNNING UNDER.
* 2 RS_SESSION                     BIT(16)
                %SESSION NUMBER OF THIS JOB
* 2 RS_PARENT_JOB_NR               BIT(16)
                %JOB NUMBER OF THE JOB THAT SPAWNED THIS ONE
* 2 RS_PRIOR_JOB_NO                BIT(16)
                %JOB NUMBER OF JOB THAT   INVOKED THIS JOB THROUGH
                %PGM.CALLER OR SORT
* 2 RS_OVLY_DESC_PTR               BIT(24)
                %ADDRESS OF RESULT DESCRIPTOR OF OVERLAY DESCRIPTOR
* 2 RS_PSEUDO_READER               BIT(24)
                %ADDRESS OF PSEUDO READER ASSIGNED TO THIS JOB
* 2 RS_DUMMY_EV                    BIT(1)
                %A GENERAL PURPOSE EVENT USED BY COOPERATING PROCESSES
                %WITHIN THE SMCP TO HANG A JOB AND CAUSE IT TO BE MOVED
                %TO THE SMCP-S COMM.QUEUE.
* 2 RS_MAX_TIME                    BIT(24)
                %IF  NEQ   0 THEN PROCESSOR TIME IN 10TH OF SECONDS THAT
                %THIS JOB IS ALLOWED TO RUN.
* 2 RS_IN_TRANSACTION              BIT(1)
                %JOB IS IN DMS TRANSACTION STATE
* 2 RS_DM_OPERATION                BIT(1)
                %JOB HAS A DMS OPERATION IN PROCESS -
                %CANNOT BE ROLLED OUT
* 2 RS_PATH_DIC                    BIT(24)
                %MEMORY ADDRESS OF DATA MGMT WORKAREA
* 2 RS_DMS_GLOBALS                 BIT(24)
                %ADDRESS OF DMS GLOBAL SPACE
```

```
, 2 RS_MFID_CHANGED                 BIT(2)
              %MUST SHIFT NAME LEFT ONE NAME BECAUSE OF USERCODE
, 2 RS_PKID_CHANGED                 BIT(1)
              %MUST DELETE THE PACK ID IN THE NAME
, 2 RS_IIO_IN_PROCESS               BIT(1)
              %INDICATES INITIALIZER I/O IS IN PROCESS
, 2 RS_MCS_FL                       BIT(8)
              %MCS FILE NUMBER FOR COBOL74 PARTICIPATING OUTPUT.
              %IS A COBOL 74 PROGRAM DOING DATA COMM
, 2 RS_TRACE_TO_BE_STOPPED          BIT(1)
              %ONE MEANS TRACE FILE WILL BE CLOSED NEXT TIME INTERP
              %DOES A WRITE TO THE TRACE FILE.
, 2 RS_CHARGE_NUMBER                BIT(24)
              %THIS JOBS CHARGE NUMBER
, 2 RS_RESTRICTIONS                 BIT(4)
              %WHETHER PRINTER,CARDS ACCESSED DIRECTLY;
              %WHETHER USERCODE IS OK OR REQUIRED IN ZIPS.
, 2 RS_BNA_ZIP                      BIT(1)
              %PROGRAM ZIPPED AN "AT" CONTROL CARD COMMAND.
  %
;%
```

PROGRAM PARAMETER BLOCK

```
DEFINE    PPB_LEVEL AS #-5#,
          PPB_SIZE AS #3743#,
          WORKING_PPB_SIZE AS #2880#,
          MASTER_PPB_SIZE AS #1440#,
          SPAD_PPB_SIZE AS #1440#,
          SCHED_SIZE AS #863#;
 %
DEFINE PPB_MASTER AS#,
  02 PROG_NAME                          CHAR(30), %
     03 PROG_CURRENT_DIRECTORY          NAME,     %
                    % DIRECTORY IN WHICH PROGRAM IS LISTED.
     03 PROG_NAME_FIRST                 NAME,     %
                    % PROGRAMS FIRST NAME
     03 PROG_NAME_SECOND                NAME,     %
                    % PROGRAMS SECOND NAME
                    %
                    % FOR COMPILATIONS - IN THE LOG COPY OF THE PPB
                    % PROG.NAME.FIRST = COMPILERS FIRST NAME
                    % PROG.NAME.SECOND = OBJECT PROGRAMS FIRST NAME
                    %
  02 PROG_INTRINSIC                   CHAR (20),
     03 PROG_INTRINSIC_DIRECTORY      NAME,
                    % PACK ID FOR INTRINSICS
     03 PROG_INTRINSIC_NAME           NAME,
                    % FAMILY NAME FOR INTRINSICS
  02 PROG_INTERP_NAME                   CHAR(30), %
     03 PROG_INTERP_DIRECTORY          NAME,     %
                    % PACK ID FOR INTERPRETER
     03 PROG_INTERP_NAME_FIRST         NAME,     %
                    % FAMILY NAME FOR INTERPRETER
     03 PROG_INTERP_NAME_SECOND        NAME,     %
                    % OFFSPRING NAME FOR INTERPRETER
  02 PROG_PRIORITY                      BIT (4),  %
                    % PRIORITY IN THE MIX - COMPILER DEFAULT = 4
  02 PROG_BEGINNING                     BIT (32), %
                    % FIRST INSTRUCTION POINTER
  02 PROG_STATIC_CORE                   WORD,     %
                    % LENGTH IN BITS OF MEMORY TO BE ALLOCATED
                    % IMMEDIATELY AFTER THE BASE REGISTER
                    % IF THERE EXISTS A DATA DICTIONARY, THEN THIS FIELD
                    % MUST BE DESCRIBED BY ITS FIRST ENTRY. IF A DISK
                    % ADDRESS IS PRESENT THERE THEN THE SPACE WILL NOT
                    % ONLY BE ALLOCATED BUT ALSO FILLED FROM THAT
                    % ADDRESS.
  02 PROG_DYNAMIC_CORE                  WORD,     %
                    % DATA OVERLAY AREA IN BITS.
```

```
  02 PROG_TOTAL_CORE                    WORD,     %
                  % SMALLEST AMOUNT OF MEMORY REQUIRED TO RUN - IN
                  % BITS.
                  % SHOULD BE EQUAL TO -
                  % PROG.STATIC.CORE + PROG.DYNAMIC.CORE +
                  % DATA DICTIONARY SIZE + FIB DICTIONARY SIZE
  02 PROG_WORKING_SET                   WORD,     %
                  % AMOUNT OF MEMORY (IN BITS) NEEDED TO
                  % RUN THE PROGRAM EFFICIENTLY
  02 PROG_DATA_DIC                      BIT (ND_SIZE),%
                  % DATA DICTIONARY
  02 PROG_SEG_DIC                       BIT (ND_SIZE),%
                  % SEGMENT DICTIONARY
  02 PROG_FPB_ADDRESS                   ADDRESS,  %
                  % RELATIVE DISK ADDRESS OF THE FIRST FPB IN THE
                  % PROGRAM FILE (RELATIVE TO THE PPB). ALL FPBS IN
                  % THE PROGRAM MUST BE CONTIGUOUS.
  02 PROG_FILES                         BIT (8),  %
                  % TOTAL NUMBER OF FILES - 255 MAX
  02 PROG_VERSION_NO                    WORD,     %
              % REVISION LEVEL OF PPB AND FPB-S
  02 PROG_OVLY_SEG                      BIT (10), %
                  % RESERVED FOR THE SDL OVERLAY HANDLER
  02 PROG_FREEZER                       BIT(1),   %
                  %REQUESTS THAT PROGRAM NOT BE RELOCATED.
  02 PROG_LINKS                         BIT(1),   %
                  % TELLS MCP WHETHER OR NOT MEMORY LINKS ARE
                  % DESIRED IN THE DYNAMIC MEMORY AREA
  02 PROG_TRACE                         BIT (8),  %
                  % TRACE FLAGS TO ENABLE TRACING FROM THE FIRST
                  % EXECUTABLE INSTRUCTION.
  02 PROG_SCHED_PRIORITY                BIT (4),  %
                  % PRIORITY FOR SCHEDULING
  02 PROG_VIRTUAL_DISK                  WORD,     %
                  % NUMBER OF DISK SEGMENTS DESIRED FOR DATA OVERLAY
                  % IF = 0 AND DATA OVERLAYS ARE REQUIRED MCP WILL
                  % USE 1000.
                  % REQUESTS THAT PROGRAM NOT BE RELOCATED
  02 PROG_IPB                           ADDRESS,  %
                  % IF THIS IS AN INTERPRETER, THEN THIS IS THE FILE
                  % RELATIVE LOCATION OF THE INTERPRETER PARAMETERS.
  02 PROG_DYNAMIC_SPACES                BIT (8),  %
              % MAX NUMBER OF SPACES TO BE ALLOCATED IN
              % PROG.DYNAMIC.CORE.
              % USED ONLY IF PROG.LINKS = 1.
  02 PROG_M_MACHINES_LOC                WORD,     %
              % FOR CHANGE STACK SIZE COMMUNICATE.
  02 PROG_NUM_M_MACHINES                BIT (8),  %
              % ONE FOR EACH CSS COMMUNICATE.
  02 PROG_SWITCHES                      BIT(40),  %
```

```
            % FOR RUN-TIME SWITCHES
  02 PROG_PERMANENT_FLAGS                BIT(4),
     03 PROG_DMS                         BOOLEAN,  %
                  % THIS PROGRAM USES DMS.
     03 PROG_INTERPRETER_CHECK_OVERRIDE BOOLEAN, %
                  % USE THE ONE I SAID.
     03 PROG_INTR_AGGR                    BIT(1),   %
                  % THIS PROG CALLS FOR INTRINSIC.AGGREGATE.
     03 PROG_PROTECTED                    BOOLEAN,%
                  % STOPS DS GT SW ST
  02 PROG_COMPILER_LEVEL                 BIT(8),   %
                  % NEW LEVEL MEANS RECOMPILE RQD.
 %
  02 PROG_PROG_PTR                       DSK_ADR,  %
                  % ABSOLUTE ADDRESS OF THE PPB ON DISK
  02 PROG_EXECUTE_TYPE                   BIT (4),  %
                  % 1 = EXECUTE
                  % 2 = COMPILE AND GO
                  % 3 = COMPILE FOR SYNTAX
                  % 4 = COMPILE TO LIBRARY
                  % 5 = COMPILE AND SAVE
                  % 6 = GO PART OF COMPILE AND GO
                  % 7 = GO PART OF COMPILE AND SAVE
                  % 8 = IPC CALL
  02 PROG_EOJ_TYPE                       BIT (4),  %
                  % 0 = NORMAL EOJ
                  % 1 = DS OR DP
                  % 2 = ERROR CONDITION IN PROGRAM
                  % 3 = ABORTED
                  % 4 = RS-ED
                  % 5 = DEATH IN FAMILY
                  % 6 = IN SCHEDULE
  02 PROG_GENERATOR_NAME                 CHAR(30), %
     03 PROG_GENERATOR_DIRECTORY         NAME,     %
                  % PACK ID FOR COMPILER
     03 PROG_GENERATOR_NAME_FIRST        NAME,     %
                  % FAMILY NAME OF COMPILER
     03 PROG_GENERATOR_NAME_SECOND       NAME,     %
                  % OFFSPRING NAME OF COMPILER
  02 PROG_DATE_COMPILED                  BIT(36)   %
                  % COMPILATION DATE -
                  %  YEAR MONTH DAY HOUR MINUTE SECOND
#;%
 %
 %
 %
DEFINE PPB_SPAD AS #,%
  02 PROG_SPAD                           BIT (S_PAD_SIZE),%
            % M-MACHINE
  02 PROG_CHARGE_NUMBER                  WORD,     %
```

```
            % CHARGE NUMBER
  02 PROG_PATH_DIC                    ADDRESS,  %
            % DATA MANAGEMENT PATH DICTIONARY ADDRESS
  02 PROG_PATH_SIZE                   BIT (20), %
            % LENGTH OF PATH DICT IN BITS
  02 PROG_NMBR_PATHS                  BIT (8), %
            % TOTAL NUMBER OF PATHS - MAX=255
  02 PROG_NMBR_INVOKES                BIT (4),  %
            % TOTAL NUMBER OF INVOKES - MAX=15
  02 PROG_COMPILER_ATTRIBUTES         BIT(80),  %
                % INTERP MUST HAVE THEM ALL.
  02 PROG_BOOLEANS                    BIT(4),   %
                % JUST LIKE IT SAYS
     03 PROG_SORT                     BOOLEAN,  %
                % 1 IF SORT PROGRAM
     03 PROG_DMS_8_C_FORMAT           BOOLEAN,  %
                % DMS COMMUNICATES USE STRUCTURE NMBRS INSTEAD
                % OF PATH DICTIONARY INDEXES
     03 PROG_DMS_GEN_SEL_EXP          BOOLEAN,  %
                % PROGRAM CAN DO GEN SEL EXP - CURRENTS SHOULD NOT
                % BE UPDATED ON NOT FOUND.
     03 PROG_TWO_SEGMENT_FPB          BOOLEAN,  %
                % CONTAINS TWO SEGMENT FPBS.
  02 PROG_SORT_DATA                   DSK_ADR,  %
                % SORT PARAMETER ADDRESS
  02 PROG_RS_EV_LIST_SIZE               BIT(4),
  02 PROG_MAX_TIME                    WORD,
                % MAXIMUM PROCESSOR TIME ALLOWED IN MINUTES
  02 PROG_6_0_NMBR_INVOKES            BIT(6),
            % EXPANDED MAXIMUM NUMBER OF DMS INVOKES - MAX = 63
  02 PROG_NAME_TABLE                    BIT(ND_SIZE),%
                % USED FOR INTRINSIC AGGREGATES FOR NAMES CHECKING.
  02 PROG_LAYOUT_TABLE_ADDRESS        BIT(16),%
            % CODEFILE-RELATIVE SEGMENT ADDRESS OF SOL LAYOUT TABLE
  02 PROG_LAYOUT_TABLE_SIZE           BIT(12),%
            % SOL LAYOUT TABLE SIZE IN SEGMENTS
  02 PROG_INTRIN_AGGR_USED            BIT(2),%
  02 PROG_MEMORY_PRIORITY             BIT(4),
            % PRIORITY FOR CODE SEGMENTS - SYSTEM DEFAULT = 4
  02 PROG_SECONDS_BEFORE_DECAY        BIT(10),
            % NUMBER OF SECONDS AFTER AN IMPORTANT SEGMENT
            % IS LAST ACCESSED BEFORE THAT SEGMENTS MEMORY PRIORITY
            % IS LOWERED.
  02 FILLER                           BIT(1),
            % NOT USED
  02 PROG_COMPILER_COMPILE_DATE       BIT(36),
            %COMPILE DATE OF THE COMPILER
  02 PROG_SPAD_SIZE                   BIT(16),
            %SIZE IN BITS OF SCRATCH PAD. MUST BE 768 FOR B1700/B1800
  02 PROG_SPAD_PTR                    BIT(24),
```

```
            %DISPLACEMENT INTO CODE FILE FOR SCRATCH PAD.
            %DISPLACEMENT IS IN SECTORS. SHOULD BE 1 FOR B1700/B1800
  02 PROG_WAIT_LENGTH                BIT(24),
            %DMS
  02 PROG_NUMBER_OF_FPBS             BIT(12),
            %NUMBER OF FILE FPBS + NUMBER OF SUB FPBS FOR I/S FILES
  02 PROG_IPC_SIZE                   BIT(16),
            %NUMBER OF ENTRIES IN THE IPC.PARAMETER.LIST
  02 PROG_IPC_PTR                    BIT(24),
            %RELATIVE DISK ADDRESS OF THE IPC.PARAMETER.LIST
  02 PROG_IPC_MAX_SEND_PARAMS        BIT(16),
            %MAX NUMBER OF PARAMETERS THIS JOB WILL SEND VIA CALL
  02 PROG_NO_SLAVE                   BIT(1),
            %JOB CANNOT RUN ON A SLAVE PROCESSOR
  02 PROG_RESTRICTIONS               BIT(4),
          % WHETHER CARDS,PRINTER OK DIRECTLY; USERCODE OK OR REQUIRED.
  02 FILLER                          BIT(156)
            %RESERVED FOR MORE COMPILER GENERATED DATA.
#;%
 %
 %
 %
DEFINE PPB_SCHED AS #,%
  02 PROG_SCHED_LINK                 DSK_ADR,  %
                  % SCHEDULE IS A LINKED LIST
  02 PROG_SCHED_PR_COPY              BIT (4),  %
                  % SCHEDULE PRIORITY
  02 PROG_SCHED_SIZE                 WORD,     %
                  % TOTAL MEMORY REQUIRED FOR THIS JOB
                  % PROG.TOTAL.CORE
  02 PROG_JOB_NUMBER                 BIT(16), %
                  % JOB NUMBER - LIVES FROM SCHEDULE TO EOJ
  02 PROG_FLAGS                      BIT(4),
     03 PROG_UNCONDITIONAL           BIT(1),   %
                  % IF TRUE THEN RUN JOB EVEN IF PREDECESSOR ABORTS.
     03 PROG_DONT_REENTER            BIT(1),   %
                  % MY SEGMENT DICTIONARY IS NON.STANDARD.
     03 PROG_NODIF                   BIT (1),%
                  % IF SPAWNED, CONTINUE IF SPAWNER GOES AWAY
     03 PROG_WAIT_OPERATOR           BOOLEAN,  %
                  % WAITING FOR OPERATOR TO FS OR RS THIS JOB
  02 PROG_EX_AFTER_NAME              CHAR(30), %
                  % THE FOLLOWING NAMES ARE USED WHEN AN EX AFTER
                  % IS ENTERED AFTER THE FIRST PROGRAM IS ALREADY
                  % IN THE MIX
     03 PROG_EX_AFTER_DIRECTORY      NAME,     %
                  % PACK ID FOR THE EX AFTER
     03 PROG_EX_AFTER_NAME_FIRST     NAME,     %
                  % FAMILY NAME OF THE EX AFTER
     03 PROG_EX_AFTER_NAME_SECOND    NAME,     %
```

```
                % OFFSPRING NAME OF THE EX AFTER
  02 PROG_SORT_JOB_NO                   BIT(24), %
                % JOB NO OF SORT INTRINSIC
  02 PROG_SPAWNER                       BIT (1),%
                % THIS PROGRAM SPAWNED ANOTHER ONE.
  02 PROG_SUBPORT_ATTR                  BIT (1),
                % PROGRAM HAS AT LEAST ONE PORT FILE WITH
                % SUBPORT ATTRIBUTES DISK SPACE ALLOCATED.
  02 PROG_JOB_ACCTING_NO                BIT(24), %
                % THIS PROGRAMS UNIQUE ACCOUTING NUMBER
  02 PROG_PRIOR_JOB_NO                  BIT(16), %
                % PROGRAM CALLERS JOB NUMBER
  02 PROG_GENERATOR_JOB_NO              BIT(16),
                % COMPILER'S JOB NUMBER, USED BY REMOTE_DECK_OPEN.
  02 PROG_SCHED_DATE                    BIT(36),  %
                % YEAR MONTH DAY HOUR MINUTE SECOND.
  02 PROG_BOJ_DATE                      BIT(36),  %
                % YEAR MONTH DAY HOUR MINUTE SECOND.
  02 PROG_RUN_UNIT                      BIT(16),
                % JOB NUMBER OF THE RUN UNIT THAT THIS JOB WILL
                % BECOME A MEMBER OF
  02 PROG_OBJ_NAME                      CHAR(30), %
     03 PROG_OBJ_DIRECTORY              NAME,     %
                % PACK ID FOR OBJECT CODE          COMPILES ONLY
     03 PROG_OBJ_NAME_FIRST             NAME,     %
                % FAMILY NAME FOR OBJECT CODE      COMPILES ONLY
     03 PROG_OBJ_NAME_SECOND            NAME,     %
                % OFFSPRING NAME FOR OBJECT CODE   COMPILES ONLY
  02 PROG_PSEUDO_READER                 ADDRESS,
                % PSEUDO.READER TABLE ADDRESS
  02 PROG_PORT_CHAN                     BIT (7),
            % PORT AND CHANNEL OF THE DISK ON WHICH
            % THE MOTHER COPY RESIDES
  02 FILLER                             FIXED,% WAS PROG_ME_FACTOR 10.0
  02 PROG_PARENT_JOB_NR                 BIT(16),%
                % JOB NR OF WHO ZIPPED THIS ONE
  02 PROG_PARENT_QUEUE                  BIT (24),%
        % PRESENCE INDICATES JOB SPAWNED. SPECIAL BOJ, EOJ AND BACKUP
        % FILE PRESENT MESSAGES WILL BE INSERTED INTO THIS QUEUE.
        % GENERAL SPO MESSAGES WILL BE PUT INTO THIS QUEUE IF
        % PROG.LOG.SPO IS ON.
  02 PROG_LOG_SPO                       BIT (1),%
  02 PROG_USER_CODE                     BIT (10),%
        % THIS WILL BE A RECORD POINTER, RELATIVE TO ZERO, INTO A DISK
        % FILE LABELED SYSTEM/USERCODES.
  02 PROG_SESSION                       BIT (16),%
        % UNIQUE NUMBER ASSIGNED BY RJE. IT IS TO BE INSERTED INTO ANY
        % MESSAGES PASSED BACK THRU THE PARENT QUEUE.
  02 PROG_PRIVILIGED                    BOOLEAN,%
                % PROGRAM RUNNING WITH A PRIVILIGED USERCODE
```

```
  02 PROG_MFID_CHGD                    BIT (2),%
  02 PROG_PKID_CHGD                    BIT (1),%
  02 PROG_OBJ_MFID_CHGD                BIT (2),%
  02 PROG_OBJ_PKID_CHGD                BIT (1)%
             % SET IF NAME CHANGED BY RJE. USED IN IDENTIFY_MIX ETC.
#;
 %
RECORD 01 PPB_RECORD BIT (PPB_SIZE)
             PPB_MASTER
             PPB_SPAD
             PPB_SCHED
             ;
RECORD 01 MASTER_PPB_RECORD BIT (MASTER_PPB_SIZE)
             PPB_MASTER
             ;
RECORD 01 SPAD_PPB_RECORD BIT (SPAD_PPB_SIZE)
             PPB_SPAD
             ;
RECORD 01 WORKING_PPB_RECORD BIT (WORKING_PPB_SIZE)
             PPB_MASTER
             PPB_SPAD
             ;
RECORD 01 SCHED_PPB_RECORD BIT (SCHED_SIZE)
             PPB_SCHED
             ;
```

FPB DECLARATIONS (RECOMMENDED DEFAULT VALUES ARE IN [ ]).

```
DEFINE FPB_SIZE AS #2096#;
 %
 %
RECORD 01 FPB_RECORD BIT(FPB_SIZE),
  02 FILE_NAME                          CHAR(10),                    %0000
            % INTERNAL FILE NAME [CANNOT BE BLANK]
  02 NAMES                              CHAR(30),                    %0080
     03 PACK_ID                         CHAR(10),
            % PACK NAME    [BLANK]
     03 MULTI_FILE_ID                   CHAR(10),
            % FAMILY NAME  [FPB.FILE_ID]
     03 FILE_ID                         CHAR(10),
            % OFFSPRING NAME  [BLANK]
        10 FILLER                       CHAR(5),       % DMS
        10 AUDITFILE_NUMBER             CHAR(5),       % DMS
  02 HDWR                               BIT(6),                      %0320
            % HARDWARE TYPE  [17]
  02 MODE                               BIT(4),                      %0326
            % RECORDING MODE
     03 EVEN_PARITY                     BIT(1),
            % 1 = EVEN        [0]
            % 0 = ODD
     03 CODE_TYPE                       BIT(3),
            % 000 = EBCDIC     [000]
            % 001 = ASCII
            % 010 = BCL
            % 011 = BINARY
  02 FILE_TYPE                          BIT(8),                      %0330
            % DISTINGUISHES DIFFERENT TYPES OF FPB-S
            % FILE TYPE TO BE USED AT CLOSE ON DISK FILES.
            % IF FPB_NEW_FORMAT EQL -3 THEN USE THIS FIELD OTHERWISE
            % USE FPB_OLD_FILE_TYPE FIELD.
            %     0 = DATA              [0]
            %     3 = PSEUDO CARD
            %     7 = INTERPRETER
            %     8 = CODE
            %     9 = DATA
            %    12 = INTRINSIC
            %    17 = RELATIVE                    (ANSI74 COBOL)
            %    18 = INDEX_SEQ_GLOBAL_FILE       (ANSI74 COBOL)
            %    19 = INDEX_SEQ_DATA_SET_FILE     (ANSI74 COBOL)
            %    20 = INDEX_SEQ_INDEX_FILE        (ANSI74 COBOL)
            % FILE TYPES 7,8 OR 12 CAUSE CLOSE TO MODIFY THE CLOSE
            % ADVERB TO CLOSE LOCK,CRUNCH
  02 BUFFERS                            BIT(16),                     %0338
            % NUMBER OF BUFFERS REQUESTED  [1]
```

```
02 BACKUP                                BIT(2),                    %0354
            % 00 = INVALID CASE
            % 01 = TAPE ONLY
            % 10 = DISK ONLY
            % 11 = EITHER TAPE OR DISK  [11]
02 BACKUP_OK                             BIT(1),                    %0356
            %SEND TO BACKUP IF NECESSRY  [1]
02 HDWR_OK                               BIT(1),                    %0357
            % SEND TO HARDWARE IF POSSIBLE [1]
02 BOOLEANS                              BIT(24),                   %0358
            %     [0]
   03 FORMS                              BIT(1),
            % OUTPUT FILE REQUIRES SPECIAL FORMS
   03 OPTIONAL                           BIT(1),
            % THIS IS AN OPTIONAL FILE
   03 VARIABLE                           BIT(1),
            % THIS FILE CONTAINS VARIABLE LENGTH RECORDS
   03 LOCK_IT                            BIT(1),
            % LOCK THIS FILE AT TERMINATE TIME
   03 COBOL_FILE                         BIT(1),
            % COBOL FILE - IMPLIED OPEN NOT ALLOWED
   03 EOP                                BIT(1),
            % INDICATES PRESENCE OF END-OF-PAGE ACTION LABELS
   03 DEFAULT                            BIT(1),
            % BLOCK.SIZE RECORD.SIZE ETC. TO BE FILLED IN BY THE MCP
            % FOR DISK FILES ONLY
   03 PSEUDO                             BIT(1),
            %FLAGS CTLDCK AND PSEUDO READER FILES
   03 RMT_KEY                            BIT(1),
            % KEY FIELD HAS BEEN ASSIGNED FOR NDL
   03 NO_LABEL                           BIT(1),
            % IF SET USE FPB.UNITNAME AND IGNORE LABEL
   03 WORK_FILE                          BIT(1),
            % FORCE JOB NO. INTO THE MIDDLE OF THE MFID
   03 QUEUE_FILE                         BIT(1),
            % IF 0 THEN SINGLE QUEUE IF 1 THEN MULTIPLE QUEUES
   03 DMS_FLAG                           BIT(1),
            % DMS AUDIT FILE
   03 LDUMP                              BIT(1),
            %THIS WILL BE A LIBRARY TAPE
   03 EMULATOR_TAPE                      BIT(1),
            % NO MCP TAPE SERVICES PROVIDED
   03 HEADER                             BIT(1),
            % USERS REMOTE READS AND WRITES WILL INCLUDE THE HEADER
   03 TRANSLATE                          BIT(1),
            % TRANSLATE ALL RECORDS USING FPB_TRANSLATE_NAME FILE
   03 USER_BACKUP_NAME                   BIT(1),
            % USE USER PRINT FILE NAME FOR NAME OF PRINTER BACKUP FILE
            % RATHER THAN MCP GENERATED NAME.
   03 ATTRIBUTE_ERROR                    BIT(1),
```

```
            %INDICATES WHETHER ERROR OCCURRED ON LAST
            % FILE ATTRIBUTE REQUEST
      03 DUMMY_FILE                    BIT(1),
            % NO LOGICAL I/O TO BE DONE ON THIS FILE
      03 COBOL74                       BIT(1),
            % ANSI74 COBOL FILE
      03 SIMPLE_HEADERS                BIT(1),
            % RMT FILE W/ HEADERS BUT NO CONTROL MESSAGES.
      03 BACKUP_FILE_EXTENTION          BIT(1),
            % EXTENTION FILE WHEN FIRST FILE EXCEEDS FILE SIZE.
      03 FLEXIBLE                      BIT(1),
            % IF EFFECT, REQUESTS A MAXIMUM OF 105 DISK AREAS.
   02 RECORD_SIZE                      BIT(24),                  %0382
            % MAX LOGICAL RECORD SIZE IN BITS
   02 RECORDS_PER_BLOCK                BIT(24),                  %0406
            % NUMBER OF LOGICAL RECORDS PER PHYSICAL RECORD  [1]
   02 MAX_BLOCK_SIZE                   BIT(24),                  %0430
            % MAX PHYSICAL RECORD SIZE IN BITS FOR VARIABLE
            % LENGTH RECORDS ONLY. NOT CONSULTED IF
            % FPB.VARIABLE = 0
            % [RECORD_SIZE * RECORDS_PER_BLOCK]
   02 ADVERB                           BIT (12),                 %0454
            %     [0]
      03 INPUT                         BIT(1),
            %
      03 OUTPUT                        BIT(1),
            %
      03 NEW                           BIT(1),
            %
      03 WITH_PUNCH                    BIT(1),
            %
      03 WITH_PRINT                    BIT(1),
            %
      03 NO_REWIND                     BIT(1),
            % DOUBLES WITH "WITH.INTERPRET".
      03 REVERSE                       BIT(1),
            % DOUBLES WITH "WITH STACKERS".
      03 OPEN_LOCK                     BIT(1),
            %
      03 OPEN_LOCKOUT                  BIT(1),
            %
      03 REPORT_NO_FILE                BIT(1),
            %
      03 REPORT_LOCKED_FILE            BIT(1),
            %
      03 OPEN_CODE                     BIT(1),
            % ADVERB USED FOR IMPLIED OPENS
   02 SAVE_UC_INDEX                    BIT(10),                  %0466
            %USED FOR BACKUP FILE EXTENTIONS
   02 FILLER                           BIT(70),                  %0476
```

```
            % 70 AVAILABLE ONES
  02 LABEL_TYPE                    BIT(4),                          %0546
            % 0 = DEFAULT ANSI LABEL  [0]
            % 1 = UNLABELED
            % 2 = 7 CHAR BURROUGHS STANDARD
  02 SAVE                          BIT(24),                         %0550
            % SAVE FACTOR FOR MAG TAPE AND DISK  [1]
  02 REEL                          BIT(24),                         %0574
            % REEL NUMBER  [1]
  02 FILLER                        BIT(24),                         %0598
            %
  02 UNITNAME                      CHAR(6),                         %0622
            % MNEMONIC OF DESIRED UNIT   [BLANK]
  02 NUMBER_STATIONS               BIT(8),                          %0670
            % NUMBER OF REMOTE STATIONS ASSIGNED TO THIS FILE
            % IF FPB_NEW_FORMAT = -2 THIS FIELD HAS BEEN REPLACED
            % BY FIRST 12 BITS OF SEGMENT 2)  [0]
  02 USE_ROUTINE                   BIT(32),                         %0678
            % SEGMENT AND DISPLACEMENT OF THE FIRST INSTRUCTION
            % IN THE USE ROUTINE      [0]
     10 DMS_AUDIT_SERIAL_NUMBER    BIT(32),          % DMS
            % SERIAL NUMBER OF FIRST AUDIT IN CURRENT AUDIT FILE
  02 USE_AREA                      BIT(48),                         %0710
            % A 24 BIT LENGTH AND 24 BIT ADDRESS
            % OF THE USE ROUTINE WORK AREA   [0]
  02 SR_STATION                    BIT(4),                          %0758
            % READ STATION FOR READER/SORTER
  02 ACCESS                        BIT(4),                          %0762
            % 0 = SERIAL              [0]
            % 1 = RANDOM
            % 6 = DELAYED_RANDOM
            % 7 RESERVED FOR EXTENDED_SEQUENTIAL_IO WHICH IS NOW
            %   IMPLEMENTED BY SETTING FPB_ACCESS = 0 AND
            %   FPB_EXTENDED_SEQUENTIAL = 1.
            % 8 = DYNAMIC
  02 AREAS                         BIT(24),                         %0766
            % MAX AREAS DESIRED
            % LIMIT = 105     [25]
  02 BLCKS_AREA                    BIT(24),                         %0790
            % NUMBER OF PHYSICAL RECORDS PER AREA  [100]
  02 EU_DRIVE                      BIT(4),                          %0814
            % SPECIAL EU OR DRIVE_NO USED WITH
            % FPB_INC_EU AND FPB_SPECIAL_EU      [0]
  02 ALL_AT_OPEN                   BIT(1),                          %0818
            % ALLOCATE ALL AREAS AT OPEN TIME  [0]
  02 PORT_KEY                      BIT(1),                          %0819
            % PROGRAM SUPPLIES SUBPORT INDEXES FOR PORT FILE
            % I/O.                          [0]
  02 MULTI_PACK_FILE               BIT(1),                          %0820
            % FILE CAN GO ON MULTI-PACK   [0]
```

```
  02 SPECIAL_EU                          BIT(1),                          %0821
             % FILE MUST GO ON EU OR DRIVE SPECIFIED BY FPB_EU_DRIVE
             % [0]
  02 INC_EU                              BIT(1),                          %0822
             % INCREMENT EU OR DRIVE FOR EACH AREA  [0]
  02 RESTORE_IMAGE                       BIT(1),                          %0823
             % RESTORE FPB IMAGE ON CLOSE RELEASE LOCK REMOVE ETC. [0]
  02 EXTENDED_SEQUENTIAL                 BIT(1),                          %0824
             % USED TO INDICATE EXTENDED_SEQUENTIAL ACCESS WHEN
             %    FPB_ACCESS = 0 OR FPB_ACCESS = 2.    [1]
  02 LINEFORMAT                          BIT(1),                          %0825
             % IF SET, WE WILL FORMAT PAGE ACCORDING TO PARAMS BELOW.
             %    [0]
  02 AUTOPRINT                           BIT(1),                          %0826
             % 0=AUTOPRINTABLE          [0]
  02 SYSTEM_BACKUP_BIT                   BIT(1),                          %0827
             % TO IDENTIFY AUTO BACKUP PROGRAM, SYSTEM/BACKUP  [0]
  02 REPETITIONS                         BIT(6),                          %0828
             % NUMBER OF COPIES OF BACKUP FILE  [0]
  02 PAGE_SIZE                           BIT(8),                          %0834
             % NUMBER OF LINES BETWEEN UPPER AND LOWER MARGINS [0]
  02 UPPER_MARGIN                        BIT(8),                          %0842
             % NUMBER OF LINES IN THE UPPER MARGIN  [0]
  02 LOWER_MARGIN                        BIT(8),                          %0850
             % NUMBER OF LINES AT THE BOTTOM OF THE PAGE [0]
  02 FOOTING                             BIT(8),                          %0858
             % LINE NUMBER FOR END-OF-PAGE (RELATIVE TO PAGESIZE) [0]
  02 OVERRIDE_MCP_CHECKS                 BIT(8),                          %0866
             %  [0]
     03 OVERRIDE_OPEN_OUTPUT_CHECK       BIT(1),
             %
  02 FILLER                              BIT(120),                        %0874
             % 120 FREE ONES
  02 MCPDATA                             BIT(36),                         %0994
             % RESERVED DISK ADDRESS (FOR MCP USE ONLY)
  02 MCPINTERNAL                         BIT(1),                          %1030
             % FOR MCP USE ONLY
  02 BACKUP_ALREADY                      BIT(1),                          %1031
             %THIS IS A BACKUP FILE - DO NOT PUT TO BACKUP AGAIN [0]
  02 NEW_FORMAT                          BIT(24),                         %1032
             %THIS WILL = -1 FOR ONE SEGMENT FPB
             %THIS WILL = -2 FOR TWO SEGMENT FPB
             %THIS WILL = -3 FOR TWO SEGMENT FPB WITH NEW FPB_FILE_TYPE
  02 OLD_FILE_TYPE                       BIT(4),                          %1056
             % IF FPB.NEW_FORMAT EQL -3 THEN THIS FIELD HAS BEEN
             % REPLACED BY FPB_FILE_TYPE
  02 PSEUDO_RDR                          BIT(24),                         %1060
             % PSEUDO READER FOR THIS FILE   [0]
  02 SAVE_HDWR                           BIT(6),                          %1084
             % SAVE HDWR TYPE IF EVER CHANGED BY OPEN  [0]
```

```
  02 INV_CHARS                          BIT(2),%                        %1090
             % 0 = REPORT ALL INVALID CHARACTER LINES
             % 1 = REPORT ALL AND STOP AT THAT LINE
             % 2 = REPORT FIRST ONE ONLY                  [2]
             % 3 = DONT REPORT ANY INVALID CHARACTERS
  02 SERIAL                             CHAR(6),%                       %1092
             %ALPHA-NUMERIC TAPE SERIAL  ID   [BLANK]
  02 Q_FAMILY_SIZE                      BIT(8),                         %1140
             % IF FPB_QUEUE_FILE IS SET, THIS FIELD IS NUMBER OF Q-S.
             % IF FPB_NEW_FORMAT = -2 THIS FIELD HAS BEEN REPLACED BY
             %   SECOND 12 BITS OF SEGMENT 2.   [0]
  02 Q_MAX_MESSAGES                     BIT(8),                         %1148
             % MAXIMUM NUMBER OF MESSAGES IN ANY ONE QUEUE
             % IF FPB_NEW_FORMAT = -2 THIS FIELD HAS BEEN REPLACED BY
             %   12 BITS IN SECOND SEGMENT OF FPB.   [0]
  02 TRANSLATE_NAME                     CHAR(10),                       %1156
             %TRANSLATE FILE NAME ("TRANSLATE"/FPB_TRANSLATE_NAME)
             %    [BLANK]
  02 PROTECTION                         BIT(2),%   HOST RJE             %1236
             % 0 = PUBLIC FILE         [0]
             % 1 = PRIVATE FILE
             % 2 = GUARD FILE
  02 PROTECTION_IO                      BIT(2),%      HOST RJE          %1238
             % 0 = I/O                  [0]
             % 1 = INPUT ONLY
             % 2 = OUTPUT ONLY
  02 MERGED_ATTRIBUTES                  BIT(12),                        %1240
             % MERGED ATTRIBUTES FROM FPB AND OPEN
             % ASSISTS IL UL AND OU AND DATA.RECORDER IO.OP GENERATION
             % CREATED BY DK.STANDARDIZE.OPEN IN OPENERS
             % FOR DATA.RECORDER.FILES ONLY AT THIS TIME
     05 FILLER                          BIT(1),
     05 MERGED_INPUTF                   BIT(1),
     05 MERGED_OUTPUTF                  BIT(1),
     05 MERGED_OPEN_OPTIONS             BIT(4),
        10 MERGED_WITH_PUNCH            BIT(1),
        10 MERGED_WITH_PRINT            BIT(1),
        10 MERGED_WITH_INTERPRET        BIT(1),
        10 MERGED_WITH_STACKERS         BIT(1),
           11 MERGED_REVERSEF           BIT(1),
     05 MERGED_LOCKF                    BIT(1),
     05 MERGED_LOCKOUTF                 BIT(1),
     05 MERGED_RRRNOFILE                BIT(1),
     05 MERGED_RNRLOCK_FILE             BIT(1),
     05 MERGED_OPENCODE                 BIT(1),
  02 MLTI_CHGD                          BIT(2),                         %1252
  02 PKID_CHGD                          BIT(1),                         %1254
  02 PROTOCOL                           BIT(8),                         %1255
             % USED BY APPLICATION PROGRAM TO TELL MCS MESSAGE FORMAT
             %     [0]
```

```
 02 EXPANDED_ADVERB                    BIT(12),                    %1263
          %      [0]
    03 INTERPRET                       BIT(1),
          %
    03 STACKERS                        BIT(1),
          % NO DOUBLE USES IN EXPANDED OPEN;
    03 IN_SECURE                       BIT(1),
          % OVERRIDE RJE NAMING CONVENTIONS AND SECURITY;
    03 OPEN_ON_BEHALF_OF               BIT(1),
          % DO NOT USE PROG'S USERCODE/PASSWORD;
    03 FILLER                          BIT(8),
          % 8 BITS AVAILABLE;
          % EXPANDED ADVERB USED ONLY BY EXPANDED OPEN;
 02 OBO_UC_INFO                        BIT(12),                    %1275
    03 OBO_UC_USED                     BIT(1),
          % OBO OPEN BIT WAS USED - REMEMBER FOR FILE CLOSE.
    03 OBO_UC_PRIV                     BIT(1),
          % FILE OPENED ON BEHALF OF PRIV USERCODE.
    03 OBO_UC_INDEX                    BIT(10),
          % USERCODE INDEX FOR WHICH FILE WAS OPENED ON BEHALF OF.
 02 ATTYPE                             BIT(16),                    %1287
      % THE ATTRIBUTE NUMBER OF THE ATTRIBUTE WHICH FAILED [0]
 02 ATTVALUE                           BIT(8),                     %1303
      % WHICH ATTRIBUTE ERROR, SEE PROCEDURE ATTVALUE FOR VALUES [0]
 02 FILLER                             BIT(129),                   %1311
          % AVAILABLE SPACE FOR GLOBAL FILE ATTRIBUTES
 02 NUMBER_STATIONS_NEW                BIT(12),                    %1440
          % IF FPB.NEW.FORMAT = -2
          % THEN NUMBER OF REMOTE STATIONS ASSIGNED TO THIS FILE [0]
 02 Q_FAMILY_SIZE_NEW                  BIT(12),                    %1452
          % IF FPB.QUEUE.FILE IS SET AND FPB.NEW.FORMAT = -2
          % THEN THIS FIELD IS NUMBER OF QUEUES.  [0]
 02 IS_SUB_FPB_PTR                     BIT(12),                    %1464
          % INDEXED/SEQUENTIAL - SUB FILE FPB OFFSET IN SEGMENTS
          %                    FROM BEGINNING OF FIRST FPB. [0]
 02 IS_NUMBER_SUB_FPBS                 BIT(8),                     %1476
          % INDEXED/SEQUENTIAL - NUMBER OF KEYS + (1 FOR DATA FILE).
 02 AUDITED                            BIT(1),                     %1484
      % DO NOT REINSTATE USER UNTIL I/O IS COMPLETE  [0]
 02 PROTECTION_ATTR                    BIT(2),                     %1485
      % HOW MUCH EXTRA EFFORT BY MCP TO SAVE FILE ?   [0]
 02 FILLER                             BIT(7),                     %1487
          %
 02 STATION_LIST_TYPE                  BIT(2),                     %1494
          %0=REMOTE FILE NAME, 1=STATION NAMES, 2=STATION LSN'S.
 02 IS_NUM_IO_DESC                     BIT(6),                     %1496
 02 AREALENGTH                         BIT(24),                    %1502
          % USED IN CONJUNCTION WITH:  FPB.RECORD_SIZE
          %                            FPB.RECORDS_BLOCK
          %                            FPB.BLOCK_SIZE
```

```
            %                                   FPB.BLOCKS_AREA
            %                                   FPB.MAX_BLOCK_SIZE
            %                                   FPB.FRAMESIZE
            % IN A VERY COMPLICATED WAY.  PREREQUISITES HAVE NOT BEEN
            % DETERMINED YET, AND WITH ANY LUCK WE MAY BE ABLE TO
            % IGNORE THIS ATTRIBUTE ALL TOGETHER.   [0]
  02 BLOCKSIZE                          BIT(24),                    %1526
            % USED IN CONJUNCTION WITH:  FPB.RECORD_SIZE
            %                           FPB.RECORDS_BLOCK
            %                           FPB.MAX_BLOCK_SIZE
            %                           FPB.FRAMESIZE
            % DEFAULTS WHEN CONFLICTS ARISE WITHIN THE DEFINITIONS OF
            % THE ABOVE ATTRIBUTES HAVE NOT YET BEEN DETERMINED. [0]
  02 DENSITY                            BIT(24),                    %1550
            % 1 = BPI200 , 2 = BPI556 , 3 = BPI800 , 4 = BPI1600
  02 FRAMESIZE                          BIT(24),                    %1574
            % THE NUMBER OF BITS TO BE TRANSFERRED AS A UNIT OF DATA.
            %     [DEFAULT ON B1700/B1800 SHOULD BE 1.]
  02 Q_MAX_MESSAGES_NEW                 BIT(12),                    %1598
            % IF FPB.NEW.FORMAT = -2 THEN THIS FIELD IS NUMBER OF
            % QUEUE MESSAGES ALLOWED IN ANY ONE QUEUE.   [0]
  02 IS_KEY_PRIME                       BIT(1),                     %1610
            % INDEXED/SEQUENTIAL - THIS FILE CONTAINS THE PRIME KEYS.
            %     [0]
  02 IS_KEY_DUP_ALLOWED                 BIT(1),                     %1611
            % INDEXED/SEQUENTIAL - MAY THIS FILE CONTAIN DUPLICATE
            %                      KEYS.    [0]
  02 IS_KEY_OFFSET                      BIT(16),                    %1612
            % INDEXED/SEQUENTIAL - BIT OFFSET IN DATA RECORD TO
            %                      BEGINNING OF THIS KEY.  [0]
  02 IS_KEY_SIZE                        BIT(12),                    %1628
            % INDEXED/SEQUENTIAL - BIT SIZE OF THIS KEY. [0]
  02 IS_KEY_SIGNED                      BIT(1),                     %1640
            % INDEXED/SEQUENTIAL - THIS FILE CONTAINS SIGNED KEYS.
            %       [0]
  02 IS_KEY_DECENDING                   BIT(1),                     %1641
            % INDEXED/SEQUENTIAL - THESE KEYS ARE IN DECENDING ORDER.
            %       [0]
  02 FILLER                             BIT(1),                     %1642
  02 SEND_ALL_ATTR                      BIT(1),                     %1643
            % TELLS HOST/SERVICES TO SEND ALL FILE ATTRIBUTES
            % TO THE COOPERATING HOST ON A BNA FILE OPEN. [0]
  02 MINRECSIZE                         BIT(24),                    %1644
            % MINIMUM RECORD SIZE FOR VARIABLE LENGTH RECORDS
            %   [5 BYTES]
  02 HOSTNAME                           CHAR(17),                   %1668
            %BNA/HOST SERVICES.   [BLANK]
  02 IS_SPLITFACTOR_PERCENT             BIT(8),                     %1804
            % PERCENT (1 - 99) OF A BLOCKS RECORDS TO KEEP IN
            % OLD BLOCK WHEN BLOCK IS SPLIT. [0]
```

```
  02 CONNECT                              BIT (8),                    %1812
               % PEER, SERVER, OR REQUESTOR PORT FILE. [0]
  02 SUBPORT_ATTR_DAD                     DSK_ADR,                    %1820
               % DISK ADDRESS OF PORT'S SUBPORT ATTRIBUTES. [0]
  02 MY_NAME                              CHAR (30)                   %1856
               % WHAT THE PROGRAM WANTS TO BE KNOWN AS ON ITS
               % PORT FILE OPEN.                    [0 OR BLANK]
;
%
%
%
```

IPB DECLARATIONS

```
DEFINE IPB_SIZE AS #1440#;
 RECORD 01 IPB_RECORD BIT(IPB_SIZE),
                02 FILLER BIT(1192),
                02 HARDWARE CHAR(1),
                02 ARCHITECTURE_NAME CHAR(10),
                02 COMPILER_LEVEL BIT(8),
                02 MCPLEVEL BIT(8),
                02 GISMO_LEVEL BIT(8),
                02 ARCHITECTURE_ATTRIBUTES BIT(80),
                02 DATA_SPACE_SIZE BIT(24),
                02 PRE_INIT_DATA_SIZE BIT(16),
                02 DATA_PTR BIT(16)
#;
 %
 %
 %
 %
 %
 %
```

HINTS

```
RECORD 1 HINTS_RECORD BIT(2256)
  2 DISPATCH_WORD                  BIT(024)%  HEX  000
  2 ADDR_OF_PROC2                  BIT(024)%  HEX  018
  2 DISK_TRACE_BLOCK_ADDR          BIT(024)%  HEX  030
  2 MASTER_GISMO                   BIT(024)%  HEX  048
  2 LOCN_MAKE_MCP_BE_HERE          BIT(033)%  HEX  060
  2 LOCN_INTERP_DICT               BIT(024)%  HEX  081
  2 KI_KO                          BIT(001)%  HEX  099
  2 NO_REINSTATES                  BIT(001)%  HEX  09A
  2 FIRE_UP_CONTROLLER             BIT(001)%  HEX  09B
  2 N_SECOND_COUNTER               BIT(002)%  HEX  09C
  2 MCP_LIMIT                      BIT(024)%  HEX  09E
  2 HINTS_LAST_ONLY                BIT(024)%  HEX  0B6
  2 MICR_DEBUG_BIT                 BIT(001)%  HEX  0CE
  2 Q_NOT_LOCKED                   BIT(001)%  HEX  0CF
  2 DFH_DIR_AD                     BIT(024)%  HEX  0D0
  2 AUTO_GUARD                     BIT(003)%  HEX  0E8
  2 FIRE_SYSTEM_BACKUP             BIT(003)%  HEX  0EB
  2 FOUND_BACKUP_DESIGNATION       BIT(001)%  HEX  0EE
  2 INTERRUPT_DISABLE_BIT          BIT(001)%  HEX  0EF
  2 TRACE_SWITCHES                 BIT(027)%  HEX  0F0
    3 FILLER                         BIT(003)%
    3 TRACE_WORD                     BIT(024)%  HEX  0F3
      4 FILLER                         BIT(011)%
      4 TRACE_CONDITIONAL_HALTS        BIT(001)%  HEX  0FE
      4 FILLER                         BIT(006)%
      4 TRACE_FLAG                     BIT(003)%  HEX  105
      4 TRACE_OPT                      BIT(003)%  HEX  108
  2 HINTS_FIRST_QUEUE              BIT(024)%  HEX  10B
  2 ADDR_OF_COLD_START_VAR         BIT(024)%  HEX  123
  2 ADDR_OF_INTERRUPT_INFO         BIT(024)%  HEX  13B
  2 MAXM                           BIT(004)%  HEX  153
  2 ELOG_FULL                      BIT(001)%  HEX  157
  2 NON_RELEASE_MCP                BIT(001)%  HEX  158
  2 GISMO_LEVEL                    BIT(008)%  HEX  159
  2 HI_RESOLUTION_TIMER_SET        BIT(001)%  HEX  161
  2 PSR_CHANGE_BIT                 BIT(001)%  HEX  162
  2 RELEASE_LEVEL                  BIT(008)%  HEX  163
  2 FIRST_LINK                     BIT(024)%  HEX  16B
  2 MCP_TYPE                       BIT(004)%  HEX  183
  2 SYCOUNTER                      BIT(020)%  HEX  187
  2 SY_CNTR_MSK                    BIT(020)%  HEX  19B
  2 SY_PRIOR_TIME                  BIT(020)%  HEX  1AF
  2 RESERVED_FOR_INTERP_USE        BIT(020)%  HEX  1C3
  2 SYSTEM_PACK_INFO               BIT(024)%  HEX  107
```

```
* 2 TRACE_PORT_CHANNEL              BIT(007)%  HEX  1EF
* 2 SYSTEM_UNIT                     BIT(012)%  HEX  1F6
*   3 SYSTEM_PORT_CHAN               BIT(007)%  HEX  1F6
*     4 SYSTEM_PORT                   BIT(003)%  HEX  1F6
*     4 SYSTEM_CHANNEL                BIT(004)%  HEX  1F9
*   3 DUMMY_BIT_RESERVED             BIT(001)%  HEX  1FD
*   3 SYSTEM_UNIT_EU                 BIT(004)%  HEX  1FE
* 2 CONSOUL_SWITCHES                BIT(028)%  HEX  202
* 2 MICRO_TRACE_FLAG                BIT(001)%  HEX  21E
* 2 GISMO_TRACE_SPACE               BIT(024)%  HEX  21F
* 2 PORT_CHANNEL_TABLE              BIT(192)%  HEX  237
* 2 BYPASS_CLEANUP                  BIT(001)%  HEX  2F7
* 2 CONTRL_CRD_FLG                  BIT(001)%  HEX  2F8
* 2 EXT_RESULT_DESC_CHAIN           BIT(024)%  HEX  2F9
* 2 T_FILES                         BIT(008)%  HEX  311
* 2 MICR_COUNT                      BIT(006)%  HEX  319
* 2 CHANGE_BIT                      BIT(001)%  HEX  31F
* 2 RELEASE_VERSION                 BIT(008)%  HEX  320
* 2 FILLER                          BIT(001)%  HEX  328
* 2 IOAT_POINTER                    BIT(024)%  HEX  329
* 2 IOAT_END                        BIT(024)%  HEX  341
* 2 SYSTEM_PAUSE_DESC               BIT(024)%  HEX  359
* 2 PSEUDO_TABLE_ADDRESS            BIT(024)%  HEX  371
* 2 DATE_SET                        BIT(001)%  HEX  389
* 2 TIME_SET                        BIT(001)%  HEX  38A
* 2 GISMO_OPTIONS                   BIT(024)%  HEX  38B
*   3 CHECK_RA                       BIT(001)%  HEX  38B
*   3 COMM_TRACE                     BIT(001)%  HEX  38C
*   3 GISMO_TRACE                    BIT(001)%  HEX  38D
*   3 XFER_24                        BIT(001)%  HEX  38E
*   3 DMS                            BIT(001)%  HEX  38F
*   3 READR_SORTR                    BIT(001)%  HEX  390
*   3 ANY_MAG_TAPE                   BIT(001)%  HEX  391
*   3 NRZ_MAG_TAPE                   BIT(001)%  HEX  392
*   3 CASSETT                        BIT(001)%  HEX  393
*   3 PAPER_TAPE                     BIT(001)%  HEX  394
*   3 DATA_COMM                      BIT(001)%  HEX  395
*   3 PORT_DEVICES                   BIT(001)%  HEX  396
*   3 EXCHANGES                      BIT(001)%  HEX  397
*   3 PORT_TRACE                     BIT(001)%  HEX  398
*   3 B1720_CODE                     BIT(001)%  HEX  399
*   3 B1860_CODE                     BIT(001)%  HEX  39A
*   3 MPROC_CODE                     BIT(001)%  HEX  39B
*   3 B1830_CODE                     BIT(001)%  HEX  39C
*   3 READ_AFTER_WRITE_CHECK         BIT(001)%  HEX  39D
*   3 PRIORITY_MEMORY_MGMT           BIT(001)%  HEX  39E
*   3 FIFO_MEMORY_MGMT               BIT(001)%  HEX  39F
*   3 THRASHING_COUNTING             BIT(001)%  HEX  3A0
*   3 FILLER                         BIT(002)% RESERVED FOR SYSTEM/INIT
* 2 DCH_SCRATCH_MEM_ADDR            BIT(024)%  HEX  3A3
```

```
* 2 S_MCP_TRACE                    BIT(001)%  HEX  3BB
* 2 INTERRUPT_SWITCH_SET           BIT(001)%  HEX  3BC
* 2 DISABLE_INTERRUPT_SW           BIT(001)%  HEX  3BD
* 2 FILLER                         BIT(013)%  HEX  3BE
* 2 BEEN_THRU_MCP_BE_HERE          BIT(001)%  HEX  3CB
* 2 REMOTE_REROUTE                 BIT(001)%  HEX  3CC
* 2 QUEUE_REROUTE                  BIT(001)%  HEX  3CD
* 2 DISK_MONITOR_GISMO             BIT(004)%  HEX  3CE
* 2 SPOLOG_NEEDS_TRANSFERING       BIT(001)%  HEX  3D2
* 2 INTERPRETER_TABLE_ADDR         BIT(036)%  HEX  3D3
* 2 SPO_PORT_CHAN                  BIT(007)%  HEX  3F7
*   3 SPO_PORT                       BIT(003)%  HEX  3F7
*   3 SPO_CHANNEL                    BIT(004)%  HEX  3FA
* 2 KEYBOARD_SPO_DESC              BIT(024)%  HEX  3FE
* 2 SPO_SQ_AD                      BIT(024)%  HEX  416
* 2 CHANNELS_NOT_PRESENT           BIT(016)%  HEX  42E
* 2 LAMP_DATA_PTR                  BIT(048)%  HEX  43E
*   3 MASTER_LAMP_DATA_PTR           BIT(024)%  HEX  43E
*   3 SLAVE_LAMP_DATA_PTR            BIT(024)%  HEX  456
* 2 SEGMENT_HALT                   BIT(004)%  HEX  46E
* 2 HALT_MASK                      BIT(024)%  HEX  472
* 2 MMCP_SEGMENT_HALT              BIT(004)%  HEX  48A
* 2 MMCP_HALT_MASK                 BIT(024)%  HEX  48E
* 2 COMPILE_TIME_OPTIONS           BIT(008)%  HEX  4A6
*   3 RELEASE_VERSION_MCP            BIT(001)%  HEX  4A6
*   3 DEBUG_OPTION                   BIT(001)%  HEX  4A7
*   3 FILLER                         BIT(006)% 6 MORE OPTIONS
* 2 CONTROL_MEMORY_SIZE            BIT(004)%  HEX  4AE
* 2 MCP_VERSION_DATE               BIT(016)%  HEX  4B2
* 2 FILLER                         BIT(007)%  HEX  4C2
* 2 CURR_INTERP_DIC_ENTRIES        BIT(005)%  HEX  4C9
* 2 DM_GLOBALS                     BIT(024)%  HEX  4CE
* 2 QUEUE_ROOT                     BIT(024)%  HEX  4E6
* 2 FILLER                         BIT(024)%  HEX  4FE
* 2 DC_CHAIN                       BIT(024)%  HEX  516
* 2 FILLER                         BIT(036)%  HEX  52E
* 2 S_C_Q_EV                       BIT(001)%  HEX  552
* 2 M_C_Q_EV                       BIT(001)%  HEX  553
* 2 S_M_Q_EV                       BIT(001)%  HEX  554
* 2 S_I_Q_EV                       BIT(001)%  HEX  555
* 2 M_M_Q_EV                       BIT(001)%  HEX  556
* 2 M_I_Q_EV                       BIT(001)%  HEX  557
* 2 M_CAUSE_LOCK                   BIT(001)%  HEX  558
* 2 M_EV_FILLER                    BIT(005)%  HEX  559
* 2 M_MCP_LR                       BIT(024)%  HEX  55E
* 2 LOCK_ADDRESS                   BIT(024)%  HEX  576
* 2 FIRST_MCP_QUEUE                BIT(024)%  HEX  58E
* 2 M_LRU_SPACE                    BIT(096)%  HEX  5A6
* 2 M_NUMBER_PAGES                 BIT(024)%  HEX  606
* 2 TRACE_ADDR                     BIT(024)%  HEX  61E
```

```
* 2 M_MCP_G_IDENT                    BIT(008)%  HEX  636
* 2 COMM_SPLITTER_ADDR               BIT(024)%  HEX  63E
* 2 COMM_SPLITTER_LENGTH             BIT(016)%  HEX  656
* 2 FIRST_RUN_UNIT                   BIT(024)%  HEX  666
* 2 INDEX_SEQ_USER_COUNT             BIT(008)%  HEX  67E
* 2 MIKES_HALT_SPACE                 BIT(096)%  HEX  686
*   3 FILLER                           BIT(048)
*   3 SMCP_HALT_NOMEM_SEQ_NO           BIT(032)%  HEX  686
*   3 FILLER                           BIT(016)
* 2 GISMO_OPTIONS_TWO                 BIT(024)%  HEX  6E6
*   3 LAMP_CPU_BASE                    BIT(001)%  HEX  6E6
*   3 FIXED_LAMP_DISPLAY               BIT(001)%  HEX  6E7
*   3 VAR_LAMP_BASE                    BIT(001)%  HEX  6E8
*   3 VAR_LAMP_CPU_OVLY                BIT(001)%  HEX  6E9
*   3 VAR_LAMP_IO                      BIT(001)%  HEX  6EA
*   3 VAR_LAMP_BARGRAPH                BIT(001)%  HEX  6EB
*   3 MEMORY_BASE_UNIT_5               BIT(001)%  HEX  6EC
*   3 FILLER                           BIT(001)%  HEX  6ED
*   3 DCPU-TIMEOUT                     BIT(001)%  HEX  6EE
*   3 FILLER                           BIT(015)%  15 MORE GISMO OPTIONS
* 2 LAST_MEM_LINK                     BIT(024)%  HEX  6FE
* 2 SMCP_CPU_PRIORITY                 BIT(024)%  HEX  716
* 2 LAMP GLOBALS                      BIT(014)   HEX  72E
*   3 LAMP_SCALE                       BIT(003)   HEX  72E
*   3 VL_ALLS                          BIT(003)   HEX  731
*     4 VL_AUCPU                         BIT(001)   HEX  731
*     4 VL-AUCOLAY                       BIT(001)   HEX  732
*     4 VL-AUDOLAY                       BIT(001)   HEX  733
*   3 FILLER                           BIT(002)   HEX  734
*   3 LAMP_OPTIONS                     BIT(004)   HEX  736
*     4 FLAMPS                           BIT(001)   HEX  736
*     4 VLAMPS                           BIT(002)   HEX  737
*       5 VLAMPS_CPU_OLAY                  BIT(001)   HEX  737
*       5 VLAMPS_IO                        BIT(001)   HEX  738
*     4 VLAMPS_BAR_GRAPH                 BIT(001)   HEX  739
*   3 VL-SMCP_OLAYF                    BIT(002)   HEX  73A
*     4 VL-SMCP_OLAY                     BIT(001)   HEX  73A
*     4 FILLER                           BIT(001)   HEX  738
* 2 JOBS_SWEEPS_BEFORE_DECAY          BIT(010)%  HEX  73C
* 2 SYSTEM_ID                         BIT(012)%  HEX  746
*   3 CPU_ID                           BIT(004)%  HEX  746
*     0=ERROR      1=B1710      2=B1720
*                  3=B1830      4=B1860
*   3 MEMORY_ID                        BIT(004)%  HEX  74A
*     0=DEFAULT   1=CORRECTABLE S-MEMORY PARITY
*   3 IO_ID                            BIT(004)%  HEX  74E
*     0=DEFAULT
* 2 ELOG_HERE                         BIT(024)%  HEX  752
* 2 CLOCK_COUNT                       BIT(004)%  HEX  76A
* 2 CHIP_TABLE_ADDRESS                BIT(024)%  HEX  76E
```

```
*  2 MIX_MEMORY_PRIORITIES          BIT(016)%  HEX  786
*  2 STOP_SCHED_INPUT               BIT(001)%  HEX  796
*  2 MSEC_DISABL_THRASH_FAULT       BIT(001)%  HEX  797
*  2 DISABLE_THRASHING_FAULT        BIT(001)%  HEX  798
*  2 MCP_VARIABLE_MEM_PRIORITY      BIT(005)%  HEX  799
*  2 FILLER                         BIT(002)%  HEX  79E
*  2 MEM_SWEEP_PENDING              BIT(001)%  HEX  7A0
*  2 SAMPLING_CLOCK                 BIT(006)%  HEX  7A1
*  2 SAMPLING_INTERVAL              BIT(006)%  HEX  7A7
*  2 MEM_SWEEP_INTERVAL             BIT(010)%  HEX  7AD
*  2 MAX_SWEEP_INTERVAL             BIT(010)%  HEX  7B7
*  2 MEM_EXTEND_COUNT               BIT(002)%  HEX  7C1
*  2 OVERLAY_COUNTER                BIT(008)%  HEX  7C3
*  2 OVERLAY_TARGET                 BIT(008)%  HEX  7CB
*  2 MCF_SWEEPS_BEFORE_DECAY        BIT(010)%  HEX  7D3
*  2 MEM_DUMP_COMPLETE              BIT(001)%  HEX  7DD
*  2 MAX_MEM_PRIORITY_IN_MIX        BIT(005)%  HEX  7DE
*  2 CONTROLLER_SCHEDULED           BIT(001)%  HEX  7E3
*  2 DCPU_ID                        BIT(002)%  HEX  7E4
*    3 MASTER                       BIT(001)%  HEX  7E4
*    3 SLAVE                        BIT(001)%  HEX  7E5
*  2 FOUNTAIN                       BIT(024)%  HEX  7E6
*  2 CLEAR_START_REQD               BIT(001)%  HEX  7FE
*  2 SCHEDULER_BLOCK_COUNT          BIT(008)%  HEX  7FF
*  2 FIRE_MCS                       BIT(001)%  HEX  807
*  2 DCPU_DATA                      BIT(024)%  HEX  808
*  2 SLAVE_PRESENT                  BIT(001)%  HEX  820
*  2 SLAVE_PORT_CHANNEL             BIT(007)%  HEX  821
*  2 DCSV_ADDRESS                   BIT(024)%  HEX  828
*  2 BNA_ADDRESS                    BIT(024)%  HEX  840
*  2 REAL_MEMORY_SIZE               BIT(024)%  HEX  858
*  2 PSEUDO_MEMORY_SIZE             BIT(024)%  HEX  870
*  2 TIME-MARK                      BIT(024)%  HEX  888
*  2 MASTER_MMCP_DATA_PTR           BIT(024)%  HEX  8A0
*  2 SLAVE_MMCP_DATA_PTR            BIT(024)%  HEX  8B8
;
DECLARE HINTS HINTS_RECORD;
 %
```

FILE TYPES FOR THE RELEASE AFTER 10.0

```
DEFINE % BEGINNING OF FILE TYPE DEFINES *********************************
 %
 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 %                                                                         %
 %          VALID FILE TYPES : DFH_FILE_TYPE, FPB_FILE_TYPE                 %
 %          -----------------------------------------------                 %
 %                                                                         %
 %     ****  ALSO SEE THE PROCEDURE "FILE_TYPE_MNEMONIC"  ****             %
 %                                                                         %
 %                           10/09/79                                      %
 %                                                                         %
 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 %                                                                         %
 %                                   SSP                          CSG      %
 %     DESCRIPTION                   CODE            ABVR         CODE     %
 %-------------------------------------------------------------------------%
 %                                                                         %
 %  S Y S T E M   F I L E S                                                %
 %                                                                         %
      LOG_FILE                      AS #001#, %     "LOG "        000     %
      DIRECTORY_FILE                AS #002#, %     "DIR "        301     %
      PSR_DECK                      AS #003#, %     "DECK"        306     %
      PRT_FILE                      AS #004#, %     "PRT "        304     %
      PCH_FILE                      AS #005#, %     "PCH "        305     %
      DUMP_TYPE_FILE                AS #006#, %     "DUMP"        000     %
      INACTIVE_LOG_FILE             AS #010#, %     "TLOG"        000     %
      INTRIN_FILE                   AS #012#, %     "INSC"        119     %
      USERCODE_FILE                 AS #016#, %     "USER"        000     %
      MCP_TEMPORARY_FILE            AS #023#, %     "TEMP"        000     %
 %                                                                         %
 %    024 <-----> 059     RESERVED FOR ADDITIONAL SYSTEM FILES             %
 %                                                                         %
 %-------------------------------------------------------------------------%
 %                                                                         %
 %  D A T A   F I L E S                                                    %
 %                                                                         %
      DATA_TYPE_FILE(X)             AS                                     %
       #((X = 0 OR X = 9 OR X = 11)  OR  (X GEQ 13 AND X LEQ  15)         %
         OR (X GEQ 17 AND X LEQ 22)  OR  (X GEQ 60 AND X LEQ 109))#,      %
      UNSPECIFIED_DATA_FILE         AS #000#, %     "DATA"        000     %
      DATA_FILE                     AS #009#, %     "DATA"        000     %
      VARIABLE_LENGTH_FILE          AS #011#, %     "VAR "        000     %
      DMS_DATA_FILE                 AS #013#, %     "DMS "        000     %
      DMS_DICTIONARY                AS #014#, %     "DMSD"        000     %
      DMS_AUDIT_FILE                AS #015#, %     "AUDT"        000     %
```

```
      RELATIVE                          AS #017#, %      "REL "        000  %
      INDEX_SEQ_GLOBAL_FILE             AS #018#, %      "IS.G"        000  %
      INDEX_SEQ_DATA_SET_FILE           AS #019#, %      "IS.D"        000  %
      INDEX_SEQ_INDEX_FILE              AS #020#, %      "IS.I"        000  %
      INDEXED_TAG_FILE                  AS #021#, %      "TAGS"        000  %
      INDEXED_DATA_FILE                 AS #022#, %      "INXD"        000  %
      SDLSYMBOL                         AS #060#, %      "SDL "        000  %
      COBOL68SYMBOL                     AS #061#, %      "C068"        004  %
      RPGSYMBOL                         AS #062#, %      "RPG "        000  %
      NDLSYMBOL                         AS #063#, %      "NDL "        012  %
      FORTRANSYMBOL                     AS #064#, %      "FOR "        010  %
      MILSYMBOL                         AS #065#, %      "MIL "        000  %
      BASICSYMBOL                       AS #066#, %      "BAS "        002  %
      UPLSYMBOL                         AS #067#, %      "UPL "        000  %
      COBOL74SYMBOL                     AS #068#, %      "C074"        000  %
      FORTRAN77SYMBOL                   AS #069#, %      "F77 "        000  %
      IBASICSYMBOL                      AS #070#, %      "IBAS"        000  %
      DASDLSYMBOL                       AS #071#, %      "DASD"        005  %
      PASCALSYMBOL                      AS #072#, %      "PASC"        000  %
      SDL2SYMBOL                        AS #073#, %      "SDL2"        000  %
      NON_NATIVEDATA                    AS #074#, %      "IDTA"        000  %
      IEASIC_INTERNAL                   AS #075#, %      "IBSU"        000  %
      FORTRAN77_UNFORMATTED             AS #076#, %      "F77U"        000  %
      COBDMS_LIB                        AS #077#, %      "CDMS"        000  %
      RPGDMS_LIB                        AS #078#, %      "RDMS"        000  %
      NETWORK_INFORMATION               AS #079#, %      "NIF "        000  %
      NDL_LIBRARY                       AS #080#, %      "NDLB"        000  %
      PASCAL_INTERCHANGE_DATA           AS #081#, %      "PIDF"        000  %
      PASCAL_MODULE_DATA                AS #082#, %      "PSMD"        000  %
%                                                                            %
%     083 <-----> 109      RESERVED FOR ADDITIONAL DATA FILES                %
%                                                                            %
%----------------------------------------------------------------------------%
%                                                                            %
%  C O D E   F I L E S                                                       %
%                                                                            %
      CODE_TYPE_FILE(X)                 AS                                   %
          #(X = 8 OR (X GEQ 110 AND X LEQ 139))#,                            %
      CODE_FILE                         AS #008#, %      "CODE"        100  %
      SDLCODE                           AS #110#, %      "SDLO"        100  %
      COBOL68CODE                       AS #111#, %      "C680"        103  %
      RPGCODE                           AS #112#, %      "RPGO"        100  %
      NDLCODE                           AS #113#, %      "NDLO"        109  %
      FORTRANCODE                       AS #114#, %      "FORO"        107  %
      MILCODE                           AS #115#, %      "MILO"        100  %
      BASICCODE                         AS #116#, %      "BASO"        102  %
      UPLCODE                           AS #117#, %      "UPLO"        100  %
      COBOL74CODE                       AS #118#, %      "C740"        100  %
      FORTRAN77CODE                     AS #119#, %      "F770"        100  %
      IEASICCODE                        AS #120#, %      "IBSO"        100  %
```

```
      PASCALCODE                    AS #121#, %       "PSCO"        100   %
      SDL2CODE                      AS #122#, %       "SD20"        100   %
      SMCPCODE                      AS #123#, %       "SMCP"        120   %
      MCSCODE                       AS #124#, %       "MCSO"        100   %
      NON_NATIVECODE                AS #125#, %       "ICDE"        100   %
      B500CODE                      AS #126#, %       "B50 "        100   %
      IBM1400CODE                   AS #127#, %       "IBMO"        100   %
%                                                                           %
%     128 <-----> 139      RESERVED FOR ADDITIONAL CODE FILES               %
%                                                                           %
%---------------------------------------------------------------------------%
%                                                                           %
%   M I C R O C O D E   F I L E S                                           %
%                                                                           %
      MICROCODE_TYPE_FILE           AS                                      %
          #(X = 7 OR (X GEQ 140 AND X LEQ 169))#,                          %
      INTERPRETER_FILE              AS #007#, %       "INTP"        130   %
      SDLINTERPRETER                AS #140#, %       "SDLI"        130   %
      COBOL68INTERPRETER            AS #141#, %       "C68I"        122   %
      RPGINTERPRETER                AS #142#, %       "RPGI"        125   %
      NDLMICROCODE                  AS #143#, %       "NDLI"        130   %
      FORTRANINTERPRETER            AS #144#, %       "FORI"        123   %
      BASICINTERPRETER              AS #145#, %       "BASI"        130   %
      UPLINTERPRETER                AS #146#, %       "UPLI"        130   %
      COBOL74INTERPRETER            AS #147#, %       "C74I"        130   %
      FORTRAN77INTERPRETER          AS #148#, %       "F77I"        130   %
      IBASICINTERPRETER             AS #149#, %       "IBSI"        130   %
      PASCALINTERPRETER             AS #150#, %       "PSCI"        130   %
      SDL2INTERPRETER               AS #151#, %       "SD2I"        130   %
      MICROMCP                      AS #152#, %       "MMCP"        130   %
      GISMO                         AS #153#, %       "GSMO"        130   %
      SYSTEM_INITIALIZER            AS #154#, %       "INIT"        130   %
      B500INTERPRETER               AS #155#, %       "B5I "        130   %
      IBM1400INTERPRETER            AS #156#  %       "IBMI"        130   %
 ;       %   END OF FILE TYPE DEFINES ********************************
%                                                                           %
%     157 <-----> 169      RESERVED FOR ADDITIONAL MICRO CODE FILES         %
%                                                                           %
%---------------------------------------------------------------------------%
%                                                                           %
%   R O O M   T O   G R O W                                                 %
%                                                                           %
%     170 <-----> 199        FUTURES                                        %
%     200 <-----> 219        RESERVED FOR DISKMAP                           %
%     220 <-----> 255        FUTURES                                        %
%                                                                           %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

HARDWARE TYPES

```
DEFINE%
    DATA_RECORDER_80        AS #01#
,   CARD_PUNCH              AS #02#%
,   N O T  U S E D          AS #03#%
,   FDC_1                   AS #04#%   ICMD
,   READER_PUNCH_PRINTER    AS #05#%
,   PAPER_TAPE_READER       AS #06#%
,   PAPER_TAPE_READER_1     AS #07#%
,   PRINTER                 AS #08#%
,   READER_SORTER_2         AS #09#%
,   READER_SORTER           AS #10#%
,   DISK_FILE               AS #11#%   ANY AVAILABLE HEAD-PER-TRACK DISK
,   DFC_1                   AS #12#%   1A,1C,SYSTEM.MEMORY (HEAD-PER-TRACK)
,   DCC_2                   AS #13#%   DISK CARTRIDGE CONTROL 2 OR 3
,   DCC_1                   AS #14#%  CARTRIDGE CONTROL 1
,   DFC_1                   AS #15#%  PACK - 225, 205, 206
,   DISK_PACK               AS #16#%  ANY AVAILABLE DISK PACK
,   DISK                    AS #17#%   ANY AVAILABLE DISK STORAGE
,   DFC_3                   AS #18#%   5-N DISK
,   READER_96               AS #19#%   96 COL READER ONLY
,   PAPER_TAPE_PUNCH        AS #20#%
,   CARD_READER             AS #21#%   80 COL READER
,   SPO                     AS #22#%
,   SPO_2                   AS #23#%   CRT SPO
,   MTC_2                   AS #24#%   TAPE NINE TRACK NRZ
,   MTC_1                   AS #25#%  TAPE SEVEN TRACK UPRIGHT DRIVE
,   MTC_3                   AS #26#%   TAPE, NINE TRACK, PE
,   TAPE                    AS #27#%   ANY AVAILABLE TAPE UNIT
,   TAPE_9                  AS #28#%   ANY AVAILABLE NINE TRACK UNIT
,   TAPE_7                  AS #MTC_1#%
,   CASSETTE                AS #30#%
,   PC_5                    AS #31#%  PRINTER CONTROL-5
,   DSC                     AS #32#%  PACK 206 207
,   PC_7                    AS #33#%  PRINTER CONTROL-7
 %
 %
,   PORT_FILE               AS #60#
,   Q_FILE                  AS #61#
,   Q_FILE_OLD              AS #62#
,   REMOTE                  AS #63#%
;
```

CSG STANDARD FILE ATTRIBUTES

The following is a list of CSG STANDARD FILE ATTRIBUTES available through B1700/B1800 MASTER CONTROL PROGRAM (MCP). Some attributes are available in subset form, tailored to the B1700/B1800 environment.

These attributes all may be accessed by system programs using the GET.ATTRIBUTE / CHANGE.ATTRIBUTE communicates. Restrictions on the accessibility of these attributes are listed in abbreviated form immediately following the attribute name and number. This information is provided in 4 elements (A, B, C, D) having the following meanings meanings:

A = attribute has meaning for these devices
B = attribute is read only, write only, or read/write
C = when may attribute be read / when may attribute be written
D = attribute type (BOOLEAN, INTEGER, STRING ETC.)

ACTIVESUBPORTS ---1035 (FIB.ACTIVE_SUBPORTS)

PORTS ONLY, READ ONLY, OPENED, INTEGER

The attribute ACTIVESUBPORTS returns to the user program the number of subports in the specified port file which are in use, including all that are in any state except closed.

ACTUALMAXMSGTEXTSIZE
---1038 (SA.ACTUAL_MAX_MSG_TEXT_SIZE)

SUBPORTS ONLY, READ ONLY, OPENED, INTEGER

The ACTUALMAXMSGTEXTSIZE attribute specifies the value of MAX_MESSAGE_TEXT_SIZE agreed to by the subports in the subport dialog.

AREAADDRESS ---1 (DFH.AREA.ADDRESS)

DISK ONLY, READ/WRITE, OPENED/OPENED, INTEGER

The attribute AREAADDRESS returns the physical disk address of an area of a disk file. The AREAADDRESS attribute requires an index, the area number, as a parameter. When the AREAADDRESS attribute is set, if the area has not been allocated, the value to be assigned to the attribute is used as an absolute address into the family member and space is allocated if available. Otherwise, an attribute error is given. Area numbers begin at zero (0).

AREAALLOCATED ---2 (DFH.AREA.ADDRESS)

DISK ONLY, READ ONLY, ANYTIME, BOOLEAN

The attribute AREAALLOCATED indicates whether or not a specific area of the associated physical file has been allocated. The AREAALLOCATED attribute requires an index, the area number, as a parameter. The SBP implementation does not allow for DUPLICATED files. Area numbers begin with zero (0).

AREALENGTH ---3 (FPB.AREALENGTH)

DISK ONLY, READ/WRITE, ANYTIME/CLOSED, INTEGER

The value of the attribute AREALENGTH is the number of FRAMESIZE units in an area of the disk file.

AREAS ---4 (FPB.AREAS)

DISK ONLY, READ/WRITE, ANYTIME/CLOSED, INTEGER

The value of the attribute AREAS is the maximum number of areas a disk file can be allocated. If AREAS is zero, the default value of twenty-five (25) is used when creating a new disk file. The maximum value is 105.

ATTERR ---5 (FPB.ATTRIBUTE.ERROR)

GENERAL, READ ONLY, ANYTIME, BOOLEAN

The attribute ATTERR returns true if the last file attribute action was in error. The ATTERR attribute is reset to false after a successful attribute action. Attribute errors are non-fatal but may be informative to the object program.

Note: a similar function may be performed by the user program by checking RS.REINSTATE.MSG.POINTER immediately following an attribute request.

ATTYPE ---1029 (FPB.ATTYPE)

GENERAL, READ ONLY, ANYTIME, INTEGER

The attribute ATTYPE returns the attribute number of the last file attribute which was incorrectly referenced.

ATTVALUE ---1030 (FPB.ATTVALUE)

GENERAL, READ ONLY, ANYTIME, INTEGER

The attribute ATTVALUE returns a value indicating the type of error found when processing the attribute request indicated by the results of an ATTYPE attribute request.

The values returned are as follows:

0 = Invalid attribute number
1 = Wrong attribute communicate
2 = File not open
3 = Invalid request for this device
4 = No file header
5 = File not open output
6 = Invalid area number
7 = Area already assigned
8 = Unable to assign area
9 = Family index request on non System disk file
10 = Requested EU non System disk
11 = File not closed
12 = Invalid change request
13 = Invalid FRAMESIZE
14 = Invalid change value
15 = Invalid PAGESIZE request

16 = File not assigned
17 = Insufficient memory for FIB
18 = Invalid NEXTRECORD request on relative file
19 = Invalid attribute length specified

AUDITED ---1030 (FPB.AUDITED)

GENERAL, READ/WRITE, ANYTIME/ANYTIME, BOOLEAN

The attribute AUDITED, when set to true, causes the user program to be held waiting until each I/O operation to this file is complete.

AVAILABLE ---6

GENERAL, READ ONLY, ANYTIME, INTEGER

The AVAILABLE attribute attempts to open a file and, when impossible reports the reason for the failure without suspending the program and requiring operator intervention. However, with the use of the AVAILABLE attribute the operator still must resolve duplicate file conditions.

When tested AVAILABLE returns:
0 = the permanent file exists but is not available (ie., the file is locked out).
1 = the file is now open and assigned to the logical file. (If the file was not previously open, it was opened.)
2 = the permanent file does not exist.
4 = unmatched serial number.
10 = no resources are available to open the file.

BACKUPFILENAME ---7 (FPB.NAMES)

PRINTER/PUNCH, READ ONLY, OPEN, STRING

The BACKUPFILENAME attribute returns the file name of the intermediate file used for the logical file.

BACKUPKIND ---8 (FPB.BACKUP)

PRINTER/PUNCH, READ/WRITE, ANYTIME/CLOSED, INTEGER

The BACKUPKIND attribute indicates the peripheral associated with a logical file as an intermediate peripheral. The I/O operations of the logical file will take place on the intermediate file but all the restrictions and capabilities of the peripheral specified by the KIND attribute will be applied to the logical file. After the logical file is closed, the intermediate file may be transferred to a peripheral as specified by the KIND attribute. The TITLE attribute specifies the title of the file when it is ultimately transferred to the peripheral specified by the KIND attribute. The file name of the intermediate file may be determined with the BACKUPFILENAME attribute.

This attribute maps directly to FPB.BACKUP with the following meanings: 00 = EITHER TAPE OR DISK
10 = DISK
80 = TAPE

BACKUPPERMITTED ---9 (FPB.BACKUP.OK FPB.HDWR.OK)

PRINTER/PUNCH, READ/WRITE, ANYTIME/CLOSED, INTEGER

The BACKUPPERMITTED attribute indicates whether an intermediate peripheral may be associated with the logical file. The values are:

0 = DONTCARE - intermediate peripheral usage allowed
1 = CONTBACKUP - no intermediate peripheral usage allowed
2 = MUSTBACKUP - intermediate peripheral usage is required

BLOCK ---10 (FIB.BLOCK.COUNT)

DISK/TAPE, READ ONLY, OPENED, INTEGER

The attribute BLOCK returns the number of the logical block referenced in the last I/O statement. The MCP returns a one relative value.

BLOCKSIZE ---11 (FPB.BLOCK.SIZE)

GENERAL, READ/WRITE, ANYTIME/CLOSED, INTEGER

The value of the BLOCKSIZE attribute is the length of a block in FRAMESIZE units. BLOCKSIZE may be set only when the file is closed. If BLOCKSIZE is less than MAXRECSIZE, it will be set to MAXRECSIZE when the file is opened. The default value of BLOCKSIZE is dependent upon the physical unit (KIND) assigned to the file and the value of the attribute MAXRECSIZE. See the discussion under the MAXRECSIZE attribute.

BLOCKSTRUCTURE ---12 (FPB.VARIABLE)

GENERAL, READ/WRITE, ANYTIME/CLOSED, INTEGER

The attribute BLOCKSTRUCTURE specifies the format of the records and the structure of the file.

The mnemonics and meanings of the BLOCKSTRUCTURE attribute are as follows:

0 = FIXED blocked or unblocked fixed-length records. This value is the default value for BLOCKSTRUCTURE.

2 = VARIABLE variable length records. The record length is contained in the first four characters of the record.

BUFFERS ---13 (FPB.BUFFERS)

GENERAL, READ/WRITE, ANYTIME/CLOSED, INTEGER

The attribute BUFFERS specifies the number of BUFFERS assigned to a file. If the number of buffers is not specified a maximum of two (2) buffers will be assigned. Only in exceptional conditions do more than two buffers add to the efficiency of the I/O operations of a file.

CONNECT ---1033 (FPB.CONNECT)

PORTS ONLY, READ/WRITE, ANYTIME/CLOSED, INTEGER

The CONNECT attribute aids in the matching of ports and subports. The default value is PEER. The possible values are:

SERVER (1): The port provides services. It must be connected to a REQUESTOR port.

REQUESTOR (2): The port only make requests for services. It must be connected to a SERVER port.

PEER (3): The port may contain any type of traffic. It must be connected to a PEER port.

CREATIONDATE ---19 (DFH.CREATION.DATE)

DISK, READ ONLY, ANYTIME, INTEGER

The attribute CREATIONDATE returns the creation date of a file. The value of the CREATIONDATE attribute is returned as an integer in the form YYDDD, where YY is the year, and DDD is the day in Julian form. When a file is created, "TODAY'S DATE" is always used as the creation date of the file.

CURRENTBLOCK ---21 (FPB.BLOCKSIZE)

GENERAL, READ ONLY, ANYTIME, INTEGER

The CURRENTBLOCK attribute returns the size in FRAMESIZE units, of the block currently in use. Normally, this value is the same as the value of the attribute BLOCKSIZE. The value of CURRENTBLOCK becomes of interest for a tape file when the system encounters a short block. Refer to the STATE attribute for more information.

DATACOMPRESSION ---1137 (SA.DATA_COMPRESSION)

PORTS/SUBPORTS ONLY, READ/WRITE, ANYTIME/ANYTIME, BOOLEAN

The DATACOMPRESSION attribute specifies whether the SUBPORT-INFORMATION-UNIT contains compressed data or not. It may be set by a using process at any time, but is subject to agreement with the remote SUBPORT. It is set false unless both subports in the subport-dialog set it true.

DENSITY ---24 (FPB.DENSITY)

TAPE ONLY, READ/WRITE, ANYTIME/CLOSED, INTEGER

The attribute DENSITY specifies the recording density of a magnetic tape file.

The mnemonics and meanings of the DENSITY attribute are as follows:

0 = BPI200
1 = BPI556
2 = BPI800
3 = BPI1600
4 = BPI6250

BPI556 is not valid for a 9-track tape. BPI1600 is valid only for phase-encoded tapes. There are two exceptions in the use of the DENSITY attribute. In the creation of a multi-file tape, the density of the first file is used for all subsequent files. In the creation of a multi-reel file, the density setting remains constant from reel to reel as long as it is valid for the tape unit. The default density value is the density setting of the tape unit selected for output files and the density at which the tape was written for input files.

DEPENDENTSPECS ---25 (FPB.DEFAULT)

GENERAL, READ/WRITE, ANYTIME/CLOSED, BOOLEAN

If the attribute DEPENDENTSPECS is true, the format of the records and the structure of the logical file are to be determined by the structure of the associated laoeled permanent file. That is to say, the attributes BLOCKSTRUCTURE, MINRECSIZE, MAXRECSIZE, BLOCKSIZE, and FRAMESIZE will be changed to agree with the values used to create the file. If no permanent file is associated with the logical file (ie. a new file is being created), or if the permanent file is UNLABELED, the attribute DEPENDENTSPECS is ignored.

DIRECTION ---26 (FPB.REVERSE)

TAPE/PAPER TAPE READER, READ/WRITE, ANYTIME/CLOSED, INTEGER

The attribute DIRECTION indicates the direction in which records will be accessed from a file.

The mnemonics of the DIRECTION attribute are as follows:

0 = FORWARD
1 = REVERSE

The default value is FORWARD.

Only BLOCKSTRUCTURE equal to FIXED files can be read in a REVERSE DIRECTION.

EXTMODE ---29 (FPB.CODE.TYPE)

GENERAL, READ/WRITE, ANYTIME/CLOSED, INTEGER

The attribute EXTMODE specifies the external or physical character size (mode) of the records in a file. The mnemonics are as follows:

0 = EBCDIC (8-bit)
1 = ASCII (8-bit)
2 = BCL (6-bit)
3 = BINARY (word mode, binary, storage unit. For card file 12 bits per column, 960 bits per record).
4 = HEX (4-bit packed decimal) not implemented

by SBP.
JCL is not a valid EXTMODE when creating a file. EXTMODE will be changed to EBCDIC in this case.

The EXTMODE attribute can be overridden by the physical mode of a permanent file or unit type.

The EXTMODE of datacom files (KIND equal to REMOTE) or console files (KIND equal to SPO) can have only EBCDIC and BINARY for their EXTMODE values.

The default value for the EXTMODE attribute is EBCDIC.

FAMILYINDEX ---30 (FPB.EU.DRIVE)

DISK ONLY, READ/WRITE, ANYTIME/ANYTIME, INTEGER

The attribute FAMILYINDEX indicates the drive or unit of systems disk upon which the areas of the associated physical file are allocated. If the FAMILYINDEX is zero (0), disk areas are allocated in the system's normal manner. Unless overridden by setting the FAMILYINDEX for a specific area, the value of the FAMILYINDEX attribute when the file is opened is used to allocate each area of the file.

The FAMILYINDEX requires an index, the area number, as a parameter.

The SBP implementation allows FAMILYINDEX to be set anytime. However, due to the non-standard nature of the multi-pack file implementation, FAMILYINDEX must be set to an element of system disk.

FAMILYNAME ---31 (FPB.PACK.ID)

DISK ONLY, READ/WRITE, ANYTIME/CLOSED, STRING

The attribute FAMILYNAME indicates the family on which the physical file is located. FAMILYNAME must be a simple identifier. The default FAMILYNAME is " " implying system disk.

NOTE: The members of a disk family must be logically equivalent devices. Mass storage devices are considered logically equivalent when they have a common file and directory structure and where the segment (or sector) is of the same length and is addressed in the same manner.

FILEKIND ---32 (FPB.FILE.TYPE)

DISK ONLY, READ/WRITE, ANYTIME/ANYTIME, INTEGER

The FILEKIND attribute describes the internal structure and/or purpose of a record of a disk file.

NOTE: Only data files are expected to be portable between systems.

FILENAME ---33 (FPB.MULTI.FILE.ID FPB.FILE.ID)

GENERAL, READ/WRITE, ANYTIME/CLOSED, STRING

The attribute FILENAME is an external file name and is used to associate a logical file with a physical or permanent file. The default FILENAME for the file is the value of the INTNAME attribute.

FILESECTION ---35 (FPB.REEL)

TAPE ONLY, READ/WRITE, ANYTIME/CLOSED, INTEGER

The attribute FILESECTION is the ISO, BSI, and ANSI file section number of the first file header label record. The initial and default FILESECTON value is one (1). The maximum value for FILESECTION is 9999. The FILESECTION attribute is file-relative, that is, the value of FILESECTION is incremented only when the file is involved in a reel switch (the data of the file requires more than one physical reel of tape) and the value of FILESECTION is reset to one (1) when the file is closed.

The FILESECTION attribute is used in permanent tape file assignment along with the attributes KIND, TITLE and, when appropriate, CYCLE and VERSION. The FILESECTION attribute is also used in automatic input reel switching.

FLEXIBLE ---36 (FPB.FLEXIBLE)

DISK ONLY, READ/WRITE, ANYTIME/CLOSED, BOOLEAN

The attribute FLEXIBLE indicates whether or not a disk file can be allocated more areas, if needed, than the number originally specified by the AREAS attribute. The setting of FLEXIBLE is ignored if the file has been crunched or is DUPLICATED or is ALLOCATE.ALL.AT.OPEN.

Setting FPB.FLEXIBLE is equivalent to requesting a maximum number of areas of 105. This requirement is due to the nature of the MCP's variable length Disk File Headers. Sufficient header space must be reserved at open to handle a maximum number of areas.

FOOTING ---1027 (FPB.FOOTING)

PRINTER ONLY, READ/WRITE, ANYTIME/ANYTIME, INTEGER

The attribute FOOTING defines the footing to be recognized by the MICRO.MCP formatting procedures when formatting an output print file.

FRAMESIZE ---38 (FPB.FRAMESIZE)

GENERAL, READ/WRITE, ANYTIME/CLOSED, INTEGER

The attribute FRAMESIZE indicates the number of bits to be transferred as a unit of data. The values of the attributes MINRECSIZE, MAXRECSIZE, BLOCKSIZE, AREALENGTH, CURRENTRECORD, and CURRENTBLOCK are expressed in FRAMESIZE units.

The native mode of operation (due to bit addressibility) is one (1) bit.

HOSTNAME ---96 (FPB.HOSTNAME)

GENERAL, READ/WRITE, ANYTIME/CLOSED, STRING

The HOSTNAME attribute specifies the logical system host name in a BNA network at which the physical file is expected to reside.

INTNAME ---42 (FPB.FILE.NAME)

GENERAL, READ/WRITE, ANYTIME/ANYTIME, STRING

The attribute INTNAME is the internal file name. File label equation is accomplished by matching the internal file name to the file name on control cards. The default internal file name is generated by the compilers. The INTNAME attribute can be programmatically changed to any simple identifier less than or equal to 10 characters (ie. A/B is not a valid internal name). When the internal name of a file is changed, label equation action is initiated using the new internal name.

KIND ---43 (FPB.HDWR)

GENERAL, READ/WRITE, ANYTIME/CLOSED, INTEGER

The KIND attribute indicates the peripheral associated with the logical file. Each logical file has exactly one value for KIND. Peripherals may be specified by two KIND's: a "specific" KIND which indicates only that peripheral and a "general" KIND which indicates a set of specific KINDs.

61 = DATA_RECORDER_80
60 = CARD_PUNCH
11 = FDC_1
62 = READER_PUNCH_PRINTER
20 = PAPER_TAPE_READER
20 = PAPER_TAPE_READER_1
70 = PRINTER
40 = READER_SORTER_2
40 = READER_SORTER
10 = DISK_FILE
10 = DFC_1
11 = DCC_2
11 = DCC_1
11 = DPC_1
11 = DISK_PACK
10 = DISK
10 = DFC_3
52 = READER_96
30 = PAPER_TAPE_PUNCH
50 = CARD_READER
90 = SPO
99 = SPO_2
82 = MTC_2
81 = MTC_1 & TAPE_7
83 = MTC_3
80 = TAPE
82 = TAPE_9
84 = CASSETTE
72 = PC_5

The default value for KIND is compiler dependent. (NOTE: The STANDARD default value is READER if the logical file is opened for input and is PRINTER if the logical file is opened for output.)

LABEL ---44 (FPB.LABEL_TYPE)

GENERAL, READ/WRITE, ANYTIME/CLOSED, INTEGER

The attribute LABEL indicates whether or not the file has label records.

When creating a LABELed file, the logical I/O subsystem writes CSG STANDARD LABELS as the beginning and ending records of the file. For a tape file, the labels are followed by a tapemark, for printer files, by a skip to channel one. When creating an UNLABELED file, the label records are not included. Punch files must be LABELED unless using Direct I/O.

The CSG STANDARD mnemonics for LABELED files are "EBCDICLABEL" and "ASCIILABEL". The mnemonics for UNLABELED files are "OMITTED" and "OMITTEDEOF". The SBP implementation differs from the STANDARD in the following way: "ASCIILABEL" and "EBCDICLABEL" are determined by looking at how the VOL1 label was written via "SN". They may not be specified at open time, or through a change attribute request on "LABEL". A label type of "OMITTEDEOF" likewise has no meaning. OMITTEDEOF information is supplied through an "FR" statement after completing a reel in response to the MCP's request for the next reel.

Meaningful values for the attribute LABEL are as follows:

0 = contains "EBCDICLABEL"
1 = contains "ASCIILABEL"
2 = contains no labels - "OMITTED"
3 = contains no labels - "OMITTEDEOF"

LASTRECORD ---46 (DFH.EOF.POINTER)

DISK ONLY, READ/WRITE, ANYTIME/ANYTIME, INTEGER

The attribute LASTRECORD returns the record number of the last record in the physical file, calculated in terms of the blocking of the logical file. LASTRECORD may not be correct during periods when the file is being expanded, because the END-OF-FILE calculations are made only when necessary, during the transition from writing to reading, for example.

No space is allocated or de-allocated because of changing the value of this attribute.

LINEFORMAT ---1024 (FPB.LINEFORMAT)

PRINTER ONLY, READ, ANYTIME/CLOSED, INTEGER

The attribute LINEFORMAT indicates whether the attributes UPPERMARGIN, LOWERMARGIN, and FOOTING should be used in the formatting of a print file.

LINENUM ---48 (FIB.LINAGE.COUNTER)

PRINTER ONLY, READ ONLY, OPENED, INTEGER

The attribute LINENUM points to the next line in the logical page to be written. LINENUM can have a value of 0-255. The maximum value should never be greater than PAGESIZE.

LOWERMARGIN ---1126 (FPB.LOWER.MARGIN)

PRINTER ONLY, READ/WRITE, ANYTIME/ANYTIME, INTEGER

The attribute LOWERMARGIN defines the lower margin to be recognized by the MICRO.MCP formatting procedures when formatting an output print file.

MAXRECSIZE ---50 (FPB.RECORD.SIZE)

GENERAL, READ/WRITE, ANYTIME/CLOSED, INTEGER

The attribute MAXRECSIZE specifies the maximum size of records in the logical file in FRAMESIZE units.

The attributes MAXRECSIZE, MINRECSIZE, BLOCKSTRUCTURE, and KIND are closely related. MAXRECSIZE must be less than or equal to BLOCKSIZE. If BLOCKSTRUCTURE is equal to FIXED, BLOCKSIZE must be a multiple of MAXRECSIZE. If MINRECSIZE is set greater than MAXRECSIZE, it will be set to MAXRECSIZE. If MAXRECSIZE is equal to zero and BLOCKSIZE is greater than zero, the value of MAXRECSIZE is set to the value of BLOCKSIZE.

MAXRECSIZE maps directly onto FPB_RECORD_SIZE.

MAXSUBPORTS --1934 (FPB.Q.FAMILY_SIZE_NEW)

PORTS ONLY, READ/WRITE, ANYTIME/CLOSED, INTEGER

The MAXSUBPORTS attribute defines the maximum number of subports that this PORT may have open at any given time.

MESSAGEQUEUELIMIT -1036

PORT/SUBPORTS ONLY, READ ONLY, ANYTIME, INTEGER

The MESSAGEQUEUELIMIT attribute indicates how many messages may be queued in any user's input subport queue.

MESSAGEQUEUESIZE --1142

PORTS/SUBPORTS, READ ONLY, ASSIGNED, INTEGER

The MESSAGEQUEUESIZE attribute returns the number of messages queued to the user's input port or subport, depending on whether the subport index is zero or non-zero.

MINRECSIZE ---51 (FPB.MINRECSIZE)

GENERAL, READ/WRITE, ANYTIME/CLOSED, INTEGER

The attribute MINRECSIZE specifies the minimum size of records in the logical file in FRAMESIZE units. If MINRECSIZE is left unset or is set to zero, a default value will be assigned when the file is opened, which depends upon the value of the attribute MAXRECSIZE. If MINRECSIZE was set greater than the value of MAXRECSIZE, it will be reset equal to MAXRECSIZE.

The minimum record size used by the logical I/O subsystem for deblocking the file is determined by taking the maximum of:

A. the logical minimum record size (MINRECSIZE),
B. the minimum used when creating the physical file,

C. and the minimum allowable record size (which is dependent upon the value of the attribute BLOCKSTRUCTURE).

Files with BLOCKSTRUCTURE other than FIXED require that the minimum record size be large enough to contain the link word or record length information. This attribute has meaning for variable length records only.

MYHOSTNAME ---1032 (DCSV.DCSV.HOSTNAME)

PORTS/SUBPORTS ONLY, READ ONLY, ANYTIME, STRING

The MYHOSTNAME attribute returns the name of the Host at which the process using the PORT is executing.

MYNAME ---1031 (FPB.MY.NAME)

PORTS ONLY, READ/WRITE, ANYTIME/CLOSED, STRING

The MYNAME attribute is the name by which a process wishes to be known when opening a port.

MYUSE ---52 (FPB.INPUT FPB.OUTPUT)

GENERAL, READ/WRITE, ANYTIME/CLOSED, INTEGER

The attribute MYUSE specifies how the file will be used. The mnemonics of the MYUSE attribute are as follows:

0 = IO - BOTH INPUT and OUTPUT
1 = IN - INPUT ONLY
2 = OUT - OUTPUT ONLY

NEWFILE ---53 (FP8.NEW)

GENERAL READ/WRITE ANYTIME/CLOSED BOOLEAN

When a file is opened and can potentially be assigned to an INPUT/OUTPUT capable device such as disk or tape the NEWFILE attribute alone will be used to decide whether or not a permanent file is desired.

NEXTRECORD ---54 (FI3.KEY)

DISK/TAPE, READ ONLY, OPENED, INTEGER

The attribute NEXTRECORD returns the current position of the file. The SPS implementation of NEXTRECORD will always return a one relative value. NEXTRECORD is invalid for Relative files.

OPEN ---55 (FIB.OPEN)

GENERAL, READ ONLY, ANYTIME, BOOLEAN

The attribute OPEN indicates whether or not the file is open.

OPENNOREWIND ---43 (FPB.NO_REWIND)

TAPE/DATA RECORDER, READ/WRITE, ANYTIME/CLOSED, BOOLEAN

The attribute OPENNOREWIND when set to true, causes the positioning of the physical tape to be the responsibiliiy of the user, and the tape is not moved by the MCP, or for a DATA RECORDER, enables the INTERPRET function.

OPENWITHPRINT ---1044 (FPB.WITH_PRINT)

DATA RECORDER, READ/WRITE, ANYTIME/CLOSED, BOOLEAN

The attribute OPENWITHPRINT, when set to true, enables the option for 80/96 column data recorders to print on the cards data other than that which is being puched into the card.

OPENWITHPUNCH ---1045 (FPB.WITH_PUNCH)

DATA RECORDER, READ/WRITE, ANYTIME/CLOSED, BOOLEAN

The attribute OPENWITHPUNCH, when set to true, enables the punch operation for 80/96 column data recorders.

OPTIONAL ---56 (FPB.OPTIONAL)

GENERAL, READ/WRITE, ANYTIME/CLOSED, BOOLEAN

The attribute OPTIONAL indicates whether or not the assignment of a permanent file is optional. If the permanent file described by the attributes TITLE, KIND, etc. is not present when the file is opened, a "NO FILE" message is sent to the operator's console and the program is suspended. If OPTIONAL is true, then the operator may respond with an "OF" system input message, and the program will proceed without a physical file assigned to the logical file.

OTHERUSE ---57 (FPB.OPEN.LOCK FPB.OPEN.LOCKOUT)

GENERAL, READ/WRITE, ANYTIME/CLOSED, INTEGER

The attribute OTHERUSE specifies how the file may be used by other programs during the time that this program has the file open. The mnemonics of the OTHERUSE attribute are as follows:

0 = IO - BOTH INPUT and OUTPUT
1 = IN - INPUT ONLY
2 = OUT - OUTPUT ONLY
3 = SECURED - NEITHER INPUT or OUTPUT

The default value is IO.

PAGESIZE ---60 (FPB.PAGE.SIZE)

PRINTER ONLY, READ/WRITE, ANYTIME/ANYTIME, INTEGER

The attribute PAGESIZE indicates the number of lines on a logical page. PAGESIZE can have a value between zero (0) and 255.

PAGESIZE cannot change from zero to non-zero or vice-versa while the file is open.

PARITY ---61 (FPB.EVEN.PARITY)

PAPER TAPE/TAPE, READ/WRITE, ANYTIME/CLOSED, INTEGER

The attribute PARITY indicates the parity used on the file. The mnemonics of the PARITY attribute are:
0 = ODD (binary or standard parity)
1 = EVEN (alpha or non-standard parity)

PARITY may be set to any value for PAPER TAPE files. PARITY may be set to EVEN for 7-TRACK MAGNETIC TAPE files only.

RECORD ---128 (FIB.KEY)

DISK/TAPE, READ ONLY, OPENED, INTEGER

The attribute RECORD returns the number of the logical record referenced in the last I/O statement. The value returned is one relative.

SAVEFACTOR ---63 (FPB.SAVE)

DISK/TAPE, READ/WRITE, ANYTIME/CLOSED, INTEGER

The attribute SAVEFACTOR indicates the expiration date of a file in terms of the number of days past the creation date.

The system does not purge a tape whose SAVEFACTOR has expired unless explicitly told to do so.

The SAVEFACTOR has no system-defined meaning for a disk file.

SECURITYTYPE ---67 (FPB.PROTECTION)

DISK or PORTS, READ/WRITE, ANYTIME/CLOSED, INTEGER

For the DISK, SECURITYTYPE attribute specifies what users, apart from the owner (creator) of a permanent disk file (as identified by the USERCODE), may access the file.

For ports, the SECURITYTYPE attribute indicates the class of users who may access the port.

Mnemonics and values of the SECURITYTYPE attribute are:
0: PRIVATE (default for ports)
1: PUBLIC (default for disk)
2: GUARDED (not supported on B1000 series systems)

SERIALNO ---70 (FPB.SERIAL)

DISK/TAPE, READ/WRITE, ANYTIME/CLOSED, STRING

The attribute SERIALNO returns the serial number of the labeled tape or base member of the disk family to which the logical file is assigned. The serial number of a tape is established by the SN system input message.

A serial number is an alphanumeric string of up to six characters, left justified in a field of blanks. The value returned by the SERIALNO attribute is a string containing the serial number in six EBCDIC characters.

STATE ---72 (FIB.LIO.FILE.STATUS)

GENERAL, READ ONLY, ASSIGNED, BOOLEAN VECTOR

The attribute STATE returns information regarding the last I/O operation performed on the file:

FIELD	MEANING
[0:1]	An error has occurred - one of the following error fields will be non-zero.
[1:1]	Boundary violation.
[2:1]	Duplicate key.
[3:1]	Sequence error.

[4:1]	A data error has occurred (size error for variable rec).
[5:1]	Invalid key.
[7:1]	Parity error.
[9:1]	End-of-file or end-of-page.
[10:1]	Short block.
[13:1]	Break on output.
[15:1]	I/O timelimit has been exceeded.
[16:1]	A security violation has been attempted.
[17:1]	No disk space for user file.
[18:1]	Port or subport not open.
[19:1]	Subport-state changed.

SUBPORTSTATE ---1041 (FIB_SUBPORT_STATE)

SUBPORTS ONLY, READ ONLY, OPEN ATTEMPTED, INTEGER

The SUBPORTSTATE attribute returns the disposition of the specified subport of the port file. Its values are:

CLOSED	1
OPENED	2
DEACTIVATED	4
SHUTDOWN-IN-PROCESS	6

TITLE ---80 (FPB.NAMES)

GENERAL, READ/WRITE, ANYTIME/CLOSED, STRING

The attribute TITLE is an external file name and is used to associate a logical file with a physical or permanent file. The default TITLE for a file is the value of the INTNAME attribute.

The TITLE attribute may be set using a string of the form
"<FILE IDENTIFIER>" ON "<FAMILY IDENTIFIER>".
This will set the FILENAME attribute to "FILE IDENTIFIER", and set the FAMILYNAME attribute to "FAMILY IDENTIFIER".

The TITLE attribute will return the name of the physical file assigned to the logical file. When a non-blank FAMILYNAME is assigned to the logical file the TITLE attribute will return a string using the "ON" syntax.

TRANSLATE ---82 (FPB.TRANSLATE)

GENERAL, READ/WRITE, ANYTIME/CLOSED, INTEGER

The attribute TRANSLATE indicates the scope of translation. The mnemonics and meanings of the TRANSLATE attribute are as follows:

2=FORCESOFT: software translation will take place using user specified table.
3=NOSOFT: no software translation.

TRANSLATING ---83 (FIB.TRANSLATE.TABLE)

GENERAL, READ ONLY, ANYTIME, BOOLEAN

The attribute TRANSLATING returns true if software translation is being performed on the records of the file.

UPDATEFILE ---85 (FPB.ACCESS)

DISK ONLY, READ/WRITE, ANYTIME/CLOSED, BOOLEAN

The UPDATEFILE attribute allows the user to explicitly indicate when a disk file is to have the update I/O accessing method.

When a disk file is open and UPDATEFILE is true, the update I/O method implies that a serial (non-keyed) write following a read will write upon the record just read (the normal serial mode writes upon the next record of the file). When the UPDATEFILE attribute is false, the update I/O method will not be used.

UPPERMARGIN ---1025 (FPB.UPPER.MARGIN)

PRINTER ONLY, READ/WRITE, ANYTIME/ANYTIME, INTEGER

The attribute UPPERMARGIN defines the upper margin to be recognized by the MICRO.MCP formatting procedures when formatting an output print file.

USEDATE ---87 (DFH.ACCESS.DATE)

DISK ONLY, READ ONLY, ANYTIME, INTEGER

The attribute USEDATE returns the date when the file was last read or written by a user program, or if it is a code file, when it was last executed. The value returned is in the Julian form YYDDD. The USEDATE of a file is unaffected by library maintenance. The value of thes attribute is not updated if the file resides on a read-only unit.

VOLUMEINDEX ---89 (FPB.REEL)

TAPE ONLY, READ/WRITE, ANYTIME/CLOSED, INTEGER

The VOLUMEINDEX attribute is the file number within a volume set. The initial and default value is one (1). The value of the VOLUMEINDEX is incremented whenever the volume set is involved in a reel switch.

Passing the Operating System a null value for the VOLUMEINDEX attribute causes the attribute to be set to one (1), its default value, and marks the logical file as not requiring specific VOLUMEINDEX checking when the logical file is assigned to a permanent file.

The VOLUMEINDEX attribute is used in permanent tape file assignment along with the attributes KIND, TITLE and, when appropriate, CYCLE and VERSION. The VOLUMEINDEX attribute is also used in automatic input reel switching.

YOURHOSTNAME ---96 (SA.YOUR_HOSTNAME)

PORTS/SUBPORTS ONLY, READ/WRITE, ANYTIME/CLOSED, STRING

The YOURHOSTNAME attribute specifies the Host in the network which contains the remote process with which this logical process wishes to communicate or is communicating using this local subport. Its default value is the value of the MYHOSTNAME attribute. This attribute may be set to null (blank or ".") for subports which may be connected to any host.

YOURNAME ---1039 (SA.YOUR_NAME)

PORTS/SUBPORTS ONLY, READ/WRITE, ANYTIME/CLOSED, STRING

The YOURNAME attribute of a subport in use by a process corresponds to the MYNAME attribute of the port with which it is communicating. Its default value is the null string ("").

YOURUSERCODE ---1040 (SA.YOUR_USERCODE)

PORTS/SUBPORTS ONLY, READ/WRITE, ANYTIME/CLOSED, STRING

The YOURUSERCODE attribute of a subport indicates the usercode that the user of the remote subport may use or is using.

APPENDIX A - DISK TRACE

GENERAL DESCRIPTION

The disk trace feature allows system processes to be monitored. The file created by the trace is accessed through the COMMUNICATE.WITH.GISMO (CWG) mechanism and is accessible to any normal-state program, MCP, Micro-MCP, interpreter or GISMO. This feature allows a very high volume of data to be recorded in a short period of time, with relatively small perturbation of system performance. For example, the trace has been used to record all S-ops executed by the MCP over a run of several hours.

For purposes of analysis, the trace file may be considered as a serial disk file. Logical I/O is done in much the same fashion, and the finished appearance is identical. However, asynchronously running processes can make entries in the file; its access is not restricted to a single process.

ACCESS

Entries in the file are made through a set of CWG operators. The CWG mechanism is reached by passing GISMO a value of E in the X register. T contains the absolute address and L contains the length of a contiguous structure of the form:

```
01  DISK.TRACE.OP.AND.RECORD
    02  DISK.TRACE.OP                BIT(8)
        03  DISK.TRACE.TYPE          BIT(4)
        03  DISK.TRACE.FUNCTION      BIT(4)
    02  FILLER                       BIT(4)
    02  DISK.TRACE.MASK              BIT(24)
    02  DISK.TRACE.RECORD            BIT(N*8)
```

DISK.TRACE.TYPE should contain 2 . See the section on Disk Trace Functions for specification of the trace function, and see the section on KT Control Cards for use of DISK.TRACE.MASK.

In SOL, the CWG S-op sets up X, T, and L, and calls GISMO. It is invoked, for example, as follows:

```
:DECLARE
     01  DISK.TRACE.OP.AND.RECORD
:,        02  DISK.TRACE.OP          BIT(8)
:,        02  RESERVED               BIT(28)
:,        02  DISK.TRACE.RECORD      CHARACTER(10)
:;
:DEFINE TRACE.WRITE S
:%
:%
:DISK.TRACE.OP = TRACE.WRITE:
:RESERVED = 0;
:DISK.TRACE.RECORD = "ABCDEF";
:COMMUNICATE.WITH.GISMO (DISK.TRACE.OP.AND.RECORD);
:%
:STOP;
:FINI;
```

Note: GISMO respects the caller's state, but no more than four A-stack cells should be in use when a CWG is invoked.

DISK TRACE FUNCTIONS

The user may specify the following functions (BITS 4-7 of CWG op):

010	Start Trace	Enable entries to disk trace file. The MCP starts the trace when it opens the file.
020	Stop Trace	Disable entries to file until the next "Start Trace" op is received.
030	Enter Time	The 48-bit value of the high-resolution timer is written in the right-most 48-bits of the data area passed by the user. The data is then entered in the file. If no timer is present, the time part of the user's data is zeroed out. If the data area passed is less than 48-bits, this data is entered in the file with no change; i.e., no time value is inserted. NOTE: The timer value is returned in the space passed by the user, as well as being entered in the file.
040	Enter Record	The record part of the CWG op is entered

in the trace file. The record is assumed to be of the length specified to the MCP at open time; otherwise, it is left justified, right truncated, or garbage filled on the right if the data passed is smaller than the declared record size.

272 Initialize Time The first 48-bits of the record part contain a value to be loaded into the high-resolution timer. The MCP initializes the timer to zero when the file is opened. This op has no effect if there is no timer on the system.

KT CONTROL CARDS

The disk trace file is opened and closed through the KT control card.

OPEN The MCP will open the file, allocate disk space, allocate and initiaize buffers and descriptors according to the user's specifications in the KT message.

Optimally, no other activity would occur on this pack while the disk trace file is in use. This will minimize contention over the arm of the drive, and so minimize the busy waits necesary for trace buffers.

The file will begin accepting records as soon as the OPEN is complete. Up to this point, disk trace ops are merely ignored.

CLOSE There are two ways of closing the trace files:

a. Explicitly, by a KT CLOSE from cards or SPO.

b. Implicitly, when GISMO detects EOF.

Caveats

The logical I/O for the file is performed by GISMO. If an entry is attempted from outside GISMO before a buffer is available, an active spin occurs until a buffer is marked ready. If entries are being made very rapidly, the timing of system processes will be perturbed.

Entries made from inside GISMO are lost if there is no buffer free. The lost entries are counted, and the first record following the lost records contains:

```
      01  MISSING.ENTRY.RECORD       BIT(48)
,       02  MISSING.ENTRY.KEY        BIT(24)
,       02  MISSING.ENTRY.COUNT      BIT(24)
```

The MISSING.ENTRY.KEY contains @FFFFFF@. This affects only entries made from inside GISMO.

Due to overhead imposed by the existerce of the disk trace file, the results from the high-resolution timer will not have true micro-second resolution. The timer is stopped whenever any disk trace function is executed, masking cut most of this overhead (specifically, waits for trace buffers are masked out). Any traces using the timer should be desigred with a skeptical eye to the overhead.

INDEX