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

COMMUNICATES

This section contains all the MCP communicates that are valid as of the Mark 9.0 release of MCPII. The Table of Contents sectionalizes them by number and the Alphabetic Index contains a named page reference for each individual communicate.

F U R M A T	BITS	
VERB	0 - 11	12
OBJECT	12 - 35	24 = FIXED
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
CT.4        ADDRESS OF KEY FOR REMOTE 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
CT.4 ADDRESS OF KEY FOR REMOTE 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 = CI.VERB 04

SORTER CONTROL

CT.VERB	04
CT.OBJECT	FILE.NUMBER
CT.ADVERB	BIT
	0-5 -
	6 POCKET SELECT
	7 STOP-FLOW
	8 BATCH-COUNT
	9 POCKET LIGHT
	10 -
	11 ENDORSE
CT.1	POCKET NUMBER

SORTER READ = CI.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) - CT.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
X	2	PURGE
	3	REMOVE
	4	CRUNCH
	5	NO REWIND
X X	6	OVERRIDE NAME CONVENTION AND SECURITY
X	7	LOCK
	8	IF NOT CLOSED
	9	ROLLOUT
	10	AUDIT SWITCH
	11	TERMINATE

220

POSITION (MICRO MCP (BACKUP FILES ONLY)) - CI.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) - CI.VERB 11

(a00B@

```
ACCESS FILE PARAMETER BLOCK (FPB)
---------------------------------
CT.VERB      11
CT.OBJECT    FILE.NUMBER
CT.ADVERB    BIT
             0-10  -
             11     0=READ
                    1=WRITE
CT.1         RECEIVING FIELD BIT LENGTH
CT.2         RECEIVING FIELD BASE RELATIVE BIT ADDRESS
```

ACCESS FILE INFORMATION BLOCK (FIB) = CT.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 = CT.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

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

@OOE@

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
0-5	- 5 UPDATE name found.
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 FILLER BIT(6), length BIT(6)
CT.2	RECEIVING FIELD BASE RELATIVE BIT ADDRESS
CT.3	BIT LENGTH OF USERCODE/PASSWORD (SHOULD BE 160) else error.
CT.4	BASE-RELATIVE ADDRESS OF USERCODE/PASSWORD

(for Co Read) Yes — Can use DESC — NO

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.

CAVEAT — Read & Write may look on different packs causing WRITE DFH failure!

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) = CT.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 = CT.VERB 22

@016@

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

@019@

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 = CT.VERB 27

@ 018 @

```
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 = CT.VERB 28

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

SORT HANDLER = CT.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

@ 01E @

```
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 = CI.VERB 34

```
CT.VERB      34
CT.OBJECT    FILE NUMBER OF RECEIVING QUEUE
             (UNUSED IF COUNT ONLY IS REQUESTED)
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 MECHANISM -- INVALID KEY
            10 = QUEUE MECHANISM -- IRRECOVERABLE IO ERROR
```

FREEZE/THAW RUN STRUCTURE = CI.VERB 35

```
FREEZE/THAW RUN STRUCTURE
-------------------------
CT.VERB      35
CT.OBJECT    BIT 0 (HIGH ORDER BIT)
             0=THAW
             1=FREEZE
```

COMPILE CARD INFORMATION - CT.VERB 36

COMPILE CARD INFORMATION

CT.VERB 36

48 BITS SDL DESCRIPTOR (WHERE TO PUT INFO) IN FORMAT :
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

174
178
179

DYNAMIC MEMORY BASE - CT.VERB 37

DYNAMIC MEMORY BASE

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

MEMORY DUMP IO 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
DESC2	BIT 49-96	MESSAGE AREA 2
QUEUE.PTR	BIT 97-106	REMOTE FILE NUMBER, STATION NUMBER, OR QUEUE-FILE-FAMILY ELEMENT NUMBER

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

@02A@

ACCESS USERCODE FILE

CT.VERB 42
DESC BIT 0-47 DESCRIPTOR TO PARAMETER LIST.

PARAMETER LIST LAYOUT

MODE	BIT (4)
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	ADD ENTRY. ALL FIELDS HAVE TO BE SUPPLIED.
5	DELETE ENTRY. USERCODE AND PASSWORD MUST BE SUPPLIED TO FIND ENTRY.
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	DELETE ALL RECORDS FOR A GIVEN USERCODE. USERCODE MUST BE PRESENT.
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)
  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.
```

Security_level (2)

Max time mins BIT (16)
BNA Hostname Character (17)

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 = CT.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 = CI.VERB 45

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

LOAD.DUMP MESSAGE = CI.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 = CI.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 = CT.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

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.

CHANGE.ATTRIBUTES = CI.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.

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 = CI.VERB 54

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)
	3-11 -
	(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 RPG TO USE FOR OPEN OF MULTI-FILE TAPE FILE.
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 = CI.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 - DCH.SCRATCH.MEM
              5 - PACK.INFO.TABLE
              6 - SPO.SQ
              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 29 = 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.
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.
```

29 = '<hostname>'

@0D@

INDEXED SEQUENTIAL POSITION = CI.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 = CI.VERB 57

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 - CI.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 = CT.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

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 - CT..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.VERE	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 REPORT.FILE.MISSING
	1 REPORT.FILE.LOCKED
	2 REPORT.EXCEPTION (SECURITY ERRORS)
	3-11 -
	(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
CT.3	OPEN STATUS - RESERVED FOR THE SMCP TO KEEP TRACK OF WHERE TO RESUME IF THE ENTIRE OPEN CANNOT BE COMPLETED.

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 - CT.VERB 69

RESERVED FOR BNA.

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.DECLARATION  AS #%
DECLARE 01 DUMMY REMAPS PACK.LABEL%
,    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 = B3500 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

```
DEFINE DISK_HEADER_DECLARATION AS #DISK_HEADER_MAP(DISK_FILE_HEADER)#,

DISK_HEADER_MAP(DISK_FILE_HEADER) AS #
DECLARE
  01 DUMMY REMAPS DISK_FILE_HEADER,
    02 DFH_AREA_ADDR_OFFSET                BIT(16),
         %OFFSET INTO THE DFH (IN BITS) FOR THE FIRST AREA ADDRESS
    02 DFH_FILE_TYPE                       BIT(8),   16
         %TYPE OF FILE DESCRIBED BY THIS HEADER
    02 DFH_SELF                            DSK_ADR,  24
         %DISK ADDRESS OF THIS HEADER
    02 DFH_NO_USERS                        BIT(8),   60
         %NUMBER OF USERS WHO HAVE THIS FILE OPENED
    02 DFH_USERS_OPEN_OUT                  BIT(4),   68
         %NUMBER OF USERS WHO HAVE THIS FILE OPENED I/O OR OUTPUT
    02 DFH_OPEN_TYPE                       BIT(4),   72
         %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),   76
```

```
      %PRE-9.0 FILE TYPES
02 DFH_PERMANENT                        BIT(4), 80
      %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 = BAD FILE - CANNOT BE MOVED BY SQUASH
      %  F = SYSTEM FILE - CANNOT REMOVE, CHANGE OR SQUASH
02 DFH_JOB_WAITING_ON_CLOSE             BIT(1), 84
      %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), 85
      %THIS FILE IS NOT IN THE DIRECTORY YET
02 FILLER                               BIT(6), 86
02 DFH_HDR_SIZE                         BIT(16), 92
      %TOTAL SIZE OF THEIS HEADER (IN BITS)
02 DFH_NO_USERS_LOCK                    BIT(4), 108
      %NUMBER OF USERS WHO HAVE THIS FILE OPENED WITH LOCK
02 DFH_RECORD_SIZE                      BIT(20), 112
      %SIZE OF THE RECORDS (IN BITS)
02 DFH_FILE_LEVEL                       BIT(4), 132
      % 0 = 8.0 AND EARLIER
      % 1 = 9.0
02 DFH_RCDS_BLOCK                       BIT(20), 136
      %NUMBER OF RECORDS PER BLOCK
02 DFH_BLOCKS_AREA                      WORD,    156
      %NUMBER OF BLOCKS PER AREA
02 DFH_SEGS_AREA                        WORD,    180
      %NUMBER OF SEGMENTS OR SECTORS PER AREA
02 DFH_AREAS_RQST                       BIT(12), 204
      %MAXIMUM NUMBER OF AREAS ALLOWED IN THIS FILE
02 DFH_AREA_CTR                         BIT(12), 216
      %CURRENT HIGH AREA NUMBER ALLOCATED
02 DFH_EOF_POINTER                      WORD,    228
      %HIGHEST RECORD NUMBER WRITTEN IN THIS FILE
02 FILLER                               BIT(4),  252
02 DFH_BPS_NO                           BIT(20), 256
      %SERIAL NUMBER OF THE BASE PACK TO WHICH THIS MULTI-PACK
      %FILE BELONGS
02 FILLER                               BIT(27), 276
02 DFH_MPF                              BIT(1),  303
      %THIS IS A MULTI-PACK FILE
02 DFH_UPDATE_DATE                      BIT(16), 304
      %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),  320
02 DFH_CREATE_TIME                      BIT(20), 324
```

```
        %TIME THE FILE WAS OPENED OUTPUT NEW.
    02 FILLER                              BIT(32), 344
    02 DFH_SAVE_FACTOR                     BIT(12), 376
        %NUMBER OF DAYS TO SAVE THIS FILE. NO SIGNIFICANCE.
    02 DFH_CREATION_DATE                   BIT(16), 388
        %JULIAN DATE OF WHEN THIS FILE WAS OPENED OUTPUT NEW.
    02 DFH_ACCESS_DATE                     BIT(16), 404
        %JULIAN DATE OF WHEN THIS FILE WAS LAST OPENED. FOR CODEFILES,
        %DATE LAST EXECUTED OR MODIFIED.
    02 FILLER                              BIT(61), 420
    02 DFH_UPDATE_VERSION                  BIT(1),  481
        %DMS USE ONLY
    02 DFH_DMS_WRITE_CONTROL,
       03 DFH_DMS_TO_BE_WRITTEN                   BIT(1),  482
        %HEADER HAS BEEN UPDATED & NEEDS TO BE WRITTEN (NAHOS CHANGED)
       03 DFH_DMS_CONTROLPOINT                    BIT(1),  483
        %CHANGED DFH - MUST BE WRITTEN OUT ON NEXT CONTROLPOINT
    02 DFH_VERSION                         BIT(36), 484
        %TIME AND DATE OF THE LAST CLOSE. I/S AND DMS ONLY.
    02 DFH_PROTECTION                      BIT(2),  520
        % 0 = PUBLIC FILE
        % 1 = PRIVATE FILE
    02 DFH_PROTECTION_IO                   BIT(2),  522
        % 0 = ACCESS MAY BE I/O
        % 1 = ACCESS MAY BE INPUT ONLY
        % 2 = ACCESS MAY BE OUTPUT ONLY
    02 DFH_USERS_RANDOM                    BIT(8),  524
        %NUMBER OF USERS WHO HAVE THIS FILE OPENED RANDOM
    02 FILLER                              BIT(8),  532
    02 DFH_MINRECSIZE                      BIT(20), 540
        %MINIMUM NUMBER OF BITS IN EACH LOGICAL RECORD
    02 DFH_MAXRECSIZE                      BIT(20); 560
        %MAXIMUM NUMBER OF BITS IN EACH LOGICAL RECORD
#;                                                  580
DEFINE
   DAD_ARRAY_SIZE AS #(36 * 105)#;


DEFINE DFH_INDEXED_DAD_DECLARATION(DAD_ARRAY) AS #
DECLARE
   01 DUMMY REMAPS DAD_ARRAY,
      02 DFH_AREA_ADDRESS          DSK_ADR,
         03 DFH_UNIT               BIT(12),
            04 DFH_PC,
               05 DFH_PORT         BIT(3),
               05 DFH_CHAN         BIT(4),
            04 DFH_SER_NO_FLAG     BOOLEAN,
            04 DFH_EU              BIT(4),
         03 DFH_ADDR               BIT(24);
#;
```

```
DEFINE DFH_SUBSCRIPTED_DAD_DECLARATION(DAD_ARRAY) AS #%
DECLARE
   01 DUMMY REMAPS DAD_ARRAY,
      02 FH_AREA_ADDRESS (105)     DSK_ADR,
         03 FH_UNIT                BIT(12),
            04 FILLER              BIT(7),
            04 FH_SER_NO_FLAG      BOOLEAN,
            04 FILLER              BIT(4),
         03 FH_ADDR                BIT(24);
#;


 %
 %
```

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 #"0"# %
,   BOV  AS #1# %
,   BOF  AS #2# %
,   EOV  AS #3# %
,   EOF AS #4# %
,   PFB  AS #5# %
,   LOST AS #7# %
;       %
DEFINE STANDARD.LABEL.DECLARATION AS # %
DECLARE 01 DUMMY REMAPS L.LABEL.RECORD  %
,   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.PB         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
; # ;
DEFINE ANSI.TAPE.DECLARATION AS #
DECLARE
     01 DUMMY REMAPS ANSI.TAPE.LABEL CHAR(80)%
,     02 ID.AND.NUMBER  CHAR(4)    %
,       03 ID           CHAR(3)    %
,       03 NUMBER       CHAR(1)    %
,     02 FILLER         CHAR (76)  %
; #; %
DEFINE VOL.HEADER.DECLARATION AS #
DECLARE%
    01 DUMMY REMAPS ANSI.TAPE.LABEL %
,     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
; #;
```

```
DEFINE HEADER1.DECLARATION AS# % USE FOR HDR1,EOV1,EOF1
DECLARE%
    01 DUMMY REMAPS ANSI.TAPE.LABEL
,     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",EOV & EOF = REAL THING
,     02 SYSTEM.CODE      CHAR(13) %
,     02 FILLER           CHAR(7)  % RFS
; #;
DEFINE HEADER2.DECLARATION AS #          %HDR2,EOV2,EOF2
DECLARE                                  %
     01 DUMMY REMAPS ANSI.TAPE.LABEL     %
,     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

```
DEFINE   CSV.SIZE       AS  #934#;
 %
DECLARE COLD.START.VARIABLES TEMPLATE BIT (CSV.SIZE);
DEFINE CSV.ONE AS #, %
    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                       DSK.ADR,
    02 INTERP.DIC.ENTRIES               WORD,
    02 CS.SIZE                          WORD,
    02 DUMP.FILE                        DSK.ADR,
    02 CSV.COLD.START.LEVEL                  BIT(24),
       03 L61.NAME.TABLE                    BIT(1),
       03 FILLER      BIT(23),
    02 MPF.TABLE                        DSK.ADR,
    02 LOG.MIX.INFO                     DSK.ADR,
    02 DISK.AVAIL                       DSK.ADR,
    02 DISK.DIRECTORY                   DSK.ADR,
    02 TEMP.TABLE                       DSK.ADR,
    02 SYSTEM.DRIVES                    BIT (16),
       03 SYSTEM.DRIVE                  BOOLEAN,
    02 AVL.TAB.DISP                     BIT (64),
    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 PWS.OPTION                    BOOLEAN,
        03 DMS.OPTION                    BOOLEAN,
        03 RFAC.OPTION                   BOOLEAN,
        03 TRMO.OPTION                   BOOLEAN,
        03 DEBUG.OPTION                  BOOLEAN,
        03 DISP.OPTION                   BOOLEAN,
        03 SPOL.OPTION                   BOOLEAN,%
        03 RMSG.OPTION                   BOOLEAN,
        03 SQRM.OPTION                   BOOLEAN,
        03 TABS.OPTION                   BOOLEAN,
        03 BREL.OPTION                   BOOLEAN,
        03 MPRI.OPTION                   BOOLEAN,
        03 THRASHING.OPTION              BOOLEAN,
        03 FLMP.OPTION                   BOOLEAN,
        03 VLCP.OPTION                   BOOLEAN,
        03 VLIO.OPTION                   BOOLEAN,
        03 BRGR.OPTION                   BOOLEAN%
#;%
DEFINE CSV.TWO AS #,%
    02 FIRST.SCHED.ENTRY                DSK.ADR,
    02 FIRST.WAITING.SCHED              DSK.ADR,
    02 GISMO.TRACE.FLAGS                WORD,
    02 SYS.PC                           BIT (28),%ARRAY OF SYS PORT,CHAN
       03 SPC                           BIT (7),
    02 MIX.LIMIT                        BIT(8)%
   ,02 SPO.Q.CLEAR.START                BIT(60)% SPO.Q.POINTER, DOWN TO B
   ,02 SPO.FLAGS                        BIT(3)%
```

```
    ,  03  SPO.DISPLAY.TIME                  BOOLEAN%
    ,  03  CRT.SPO.TRANSL                    BOOLEAN%
    ,  03  CRT.SPO.DIRN                      BOOLEAN%
    ,02 SPOLOG.LAST.AREA                   BIT(1)% SPO.LOG.FULL
    ,02  SPO.SUPPRESS                      BIT(1)% IF ON THE DISPLAY ONLY IO
    ,02  FILLER                            BIT(1)%
    ,02  SPOLOG.LAST.SECTORS               BIT(2)% NOT USED TILL 6.1
    ,02  SPOLOG.SIZE                       BIT(16)% 14 BITS ARE ENOUGH.,
    ,02 NEXT.SPOLOG.REC                    BIT(36) %NEXT SPOLOG SEGMENT,
,    02 CSV.CORRECTABLE.ERROR.TABLE.LENGTH   BIT(16)%  40 + 32*#ENTRIES
    ,02 DUMP.FILE.SIZE                       BIT(16)
#;%,
DEFINE CSV.DECLARATION AS #%
DECLARE 01 DUMMY REMAPS COLD.START.VARIABLES
CSV.ONE
CSV.TWO
;#;%
DEFINE DCSV.SIZE AS #1071#;
 %
DEFINE DCSV.DECLARATION AS #%
DECLARE 01 DUMMY REMAPS DISK.CS.VARIABLES,
    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 CTLOCK.NO                         WORD,
    02 LOG.NO                            WORD,
    02 Q.DISK                            DSK.ADR,
    02 TRACE.FPB                         DSK.ADR,
    02 AUTO.MASK                         BIT(28),% 4 OF 7 EACH=PRT PC
    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),
         %ARRAY OF 16 16 BIT ENTRIES OF UNITS TO BE PROTECTED THRU C/S
         %12 BITS FOR PCU
         % 2 BITS FOR LABEL TYPE
         %    0 = ANSII
         %    1 = UNLABELED
         %    2 = BURROUGHS
```

```
          % 1 BIT FOR TRANSLATE  0=EBCDIC  1=ASCII
          % 1 BIT TO INDICATE THAT A C/S HAS HAPPENED ON THIS UNIT
      02 SYS.LOG.NUMBER                    BIT(24),
      02 JOE.ACCTING.NUMBER                BIT(24),
      02 SESSION.NM                        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.
      ;
#;%
  %
  %
  %
#;%
  %
  %
  %
```

```
DEFINE  MEMORY.LINK.SIZE AS #179#;
DECLARE MEMORY.LINK TEMPLATE BIT(MEMORY.LINK.SIZE);
DEFINE MEMORY.LINK.DECLARATION AS #
DECLARE 01 DUMMY REMAPS MEMORY.LINK,
            2 ML.DISK                       DSK.ADR,
            2 ML.GROUP,
              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(22),
              3 ML.DK.INTERVAL              BIT(6),
              3 ML.CURRENT.DK.INT           BIT(6),
              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);#;

USE( ML.DISK
    ,ML.POINTER
    ,ML.JOB.NUMBER
    ,ML.TYPE
    ,ML.SAVE
    ,ML.SIZE
    ,ML.FRONT
    ,ML.BACK
     ) OF MEMORY.LINK.DECLARATION;

DEFINE Q.ML.DECLARATION AS#DECLARE
   01 Q.MEMORY.LINK TEMPLATE
,     02 FILLER                 BIT(MEMORY.LINK.SIZE)
,     02 Q.ML.F.AVL             ADDRESS
,     02 Q.ML.B.AVL             ADDRESS
;#;

DEFINE
   TAKE.LO                      AS#0#
,  TAKE.RIGHTMOST               AS#1#
;
 %
 %
 %
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);
 %
DEFINE SY.DECLARATION AS #SY.DECL(SYSTEM.DESCRIPTOR)#;%
DEFINE SY.DECL(X) AS #DECLARE%
    01  DUMMY REMAPS X,%
       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.OK.FACTOR         BIT(3)    % MEMORY DECAY FACTOR
       02 SY.SEG.PG            BIT(7),   % 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 #59#;
 %
 %
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.OS.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#
,    NOT USED                   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#%
,    NC.CONTROLLER              AS #35#%
,    NC.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#
;
 %
 %
 %
```

RUN STRUCTURE NUCLEUS

```
DEFINE RS.N.SIZE AS #2355#;
 %
DECLARE RS.NUCLEUS TEMPLATE BIT(RS.N.SIZE);
 %
DEFINE RS.INTERP.PART AS #%
 %
 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 % 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.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
 %
 %
#;
 %
DEFINE RS.GISMO.PART AS #%
 %
 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 % THE FOLLOWING ARE USED BY GISMO, MMCP AND THE SMCP
 %
```

```
* 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 @17532
* 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).
 %
#;
 %
DEFINE RS.MMCP.PART AS #%
 %
 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 %THE FOLLOWING ARE USED BY THE MMCP AND THE SMCP
 %
, 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
 %
 %
#;
 %
DEFINE RS.SMCP.PART AS #%
 %
 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 %THE FOLLOWING ARE USED ONLY BY THE SMCP
 %
, 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. INCEX INTO THE
                 %FIB.DICTIONARY
, 2 RS.SER.NO                         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 HAS 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 COBOL74 PROGRAM DOING DATA COMM.
, 2 RS.TRACE.TO.BE.STOPPED          BIT(1)
                %ONE MEANS TRACE WILL BE STOPPED NEXT
                %TIME THE INTERPRETER DOES A WRITE TO THE TRACE FILE.
, 2 RS.CHARGE.NUMBER                BIT(24)
                %THIS JOBS CHARGE NUMBER.
 %
#;
 %
 %
DEFINE RS.N.DECLARATION AS #%
DECLARE 01 DUMMY REMAPS RS.NUCLEUS
RS.INTERP.PART
RS.GISMO.PART
RS.MMCP.PART
RS.SMCP.PART
;#;%
 %
```

%
%
%
%
%

PROGRAM PARAMETER BLOCK

```
DEFINE PPB.LEVEL AS #-4#;
DEFINE PPB.SIZE AS #3870#;
DEFINE WORKING.PPB.SIZE AS #2880#;
 %
DEFINE PPB.DECLARATION AS#
DECLARE 01 DUMMY REMAPS PPB
PPB.MASTER
PPB.SPAD
PPB.SCHED
;#;
 %
 %
 %
DEFINE MASTER.PPB.DECLARATION AS#
DECLARE 01 DUMMY REMAPS MASTER.PPB
PPB.MASTER
;#;
 %
 %
 %
DEFINE SCHED.DECLARATION AS#
DECLARE 01 DUMMY REMAPS SCHED
PPB.SCHED
;#;
 %
 %
 %
DEFINE WORKING.PPB.DECLARATION AS #
DECLARE 01 DUMMY REMAPS WORKING.PPB
PPB.MASTER
PPB.SPAD
;#;%
 %
 %
 %
DEFINE MASTER.PPB.SIZE AS #1440#;
 %
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 SOL 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,
     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
  02 PROG.EOJ.TYPE                      BIT (4),  %
                    % 0 = NORMAL EOJ
                    % 1 = DS OR DP
                    % 2 = ERROR CONDITION IN PROGRAM
                    % 3 = ABORTED
                    % 4 = RS-ED
  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.8.0.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
   2 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),
              % EXPANCED 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 SDL LAYOUT TABLE
  02 PROG.LAYOUT.TABLE.SIZE           BIT(12),%
              % SDL 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),
  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),
  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 FILLER                              BIT(160)
                %RESERVED FOR MORE COMPILER GENERATED DATA.
#;%
 %
 %
 %
 %
 %
 %
 %
 %
 %
DEFINE SCHED.SIZE AS #957#;%
 %
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 FILLER                             BIT(8), %
  02 PROG.FLAGS,                                 %
     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.REMOTE.JOB.STN                BIT(8),   %
                % STATION NO. OF RJE JOB.
  02 PROG.REMOTE.JOB                    BOOLEAN,  %
                % FLAGS RJE (REMOTE) JOB
    02 PROG.SORT.JOB.NO                 BIT(24), %
```

```
                    % JOB NO OF SORT INTRINSIC
  02 PROG.SPAWNER                   BIT (1),%
                    % THIS PROGRAM SPAWNED ANOTHER ONE.
  02 FILLER                         BIT(2),
                    % 2 BITS AVAILABLE
  02 PROG.JOB.ACCTING.NO            BIT(24), %
                    % THIS PROGRAMS UNIQUE ACCOUTING NUMBER
  02 FILLER                         BIT(8),
  02 PROG.PRIOR.JOB.NO              BIT(16), %
                    % PROGRAM CALLERS JOB NUMBER
  02 FILLER                         BIT(8), %
  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 FILLER                         BIT(68),  %
                    % 68 AVAILABLE
  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 PROG.ME.FACTOR          FIXED %
, 02 PROG.PARENT.JOB.NR             BIT(16) %
                    % JOB NR OF WHO ZIPPED THIS ONE
, 02 FILLER                         BIT(8) %
,   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 PROC.OBJ.MFID.CHGD                BIT (2)%
    , 02 PROC.OBJ.PKID.CHGD                BIT (1)%
              % SET IF NAME CHANGED RY RJE. USED IN IDENTIFY MIX AND ECT
    , 02 FILLER                            BIT (9)%
#;
 %
 %
 %
 %
 %
 %
```

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

```
DEFINE FPB.SIZE AS (#1804#;)
 %
DEFINE FPB.DECLARATION AS #%
DECLARE 01 DUMMY REMAPS FPB,
  02 FPB.FILE.NAME                         NAME,                          %0000
              % INTERNAL FILE NAME  [CANNOT BE BLANK]
  02 FPB.NAMES                             CHAR (30),                     %0080
     03 FPB.PACK.ID                        NAME,
              % PACK NAME  [BLANK]
     03 FPB.MULTI.FILE.ID                  NAME,
              % FAMILY NAME  [FPB.FILE.NAME]
     03 FPB.FILE.ID                        NAME,
              % OFFSPRING NAME  [BLANK]
        10 FILLER                          CHARACTER(5),        % DMS
        10 FPB.AUDITFILE.NUMBER            CHARACTER(5),        % DMS
  02 FPB.HDWR                              BIT(6),                        %0320
              % HARDWARE TYPE
  02 FPB.MODE                              BIT (4),                       %0326
              % RECORDING MODE
     03 FPB.EVEN.PARITY                    BOOLEAN,
              % 1 = EVEN
              % 0 = ODD  [0 is the default]
     03 FPB.CODE.TYPE                      BIT (3),
              % 000 = EBCDIC  [000 is the default]
              % 001 = ASCII
              % 010 = BCL
              % 011 = BINARY
  02 FPB.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
```

```
            %     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 FPB.BUFFERS                    BIT(16),                    %0338
            % NUMBER OF BUFFERS REQUESTED  [1]
  02 FPB.BACKUP                     BIT (2),                    %0354
            % 00 = INVALIC CASE
            % 01 = TAPE ONLY
            % 10 = DISK ONLY
            % 11 = EITHER TAPE OR DISK  [11 is the default]
  02 FPB.BACKUP.OK                  BOOLEAN,                    %0356
            %SEND TO BACKUP IF NECESSRY  [1]
  02 FPB.HDWR.OK                    BOOLEAN,                    %0357
            % SEND TO HARDWARE IF POSSIBLE  [1]
  02 FPB.BOOLEANS                   BIT (24),                   %0358
     03 FPB.FORMS                   BOOLEAN,
            % OUTPUT FILE REQUIRES SPECIAL FORMS  [0]
     03 FPB.OPTIONAL                BOOLEAN,
            % THIS IS AN OPTIONAL FILE  [0]
     03 FPB.VARIABLE                BOOLEAN,
            % THIS FILE CONTAINS VARIABLE LENGTH RECORDS  [0]
     03 FPB.LOCK                    BOOLEAN,
            % LOCK THIS FILE AT TERMINATE TIME  [0]
     03 FPB.COBOL                   BOOLEAN,
            % COBOL FILE - IMPLIED OPEN NOT ALLOWED  [0]
     03 FPB.EOP                     BOOLEAN,
            % INDICATES PRESENCE OF END-OF-PAGE ACTION LABELS  [0]
     03 FPB.DEFAULT                 BOOLEAN,
            % BLOCK.SIZE RECORD.SIZE ETC. TO BE FILLED IN BY THE MCP
            % FOR DISK FILES AND TAPE FILES.  [0]
     03 FPB.PSEUDO                  BOOLEAN,
            %FLAGS CTLOCK AND PSEUDO READER FILES  [0]
     03 FPB.RMT.KEY                 BOOLEAN,
            % KEY FIELD HAS BEEN ASSIGNED FOR NDL  [0]
     03 FPB.NO.LABEL                BOOLEAN,
            % IF SET USE FPB.UNIT.NAME AND IGNORE LABEL  [0]
     03 FPB.WORK.FILE               BOOLEAN, %
            % FORCE JOB NO. INTO THE MIDDLE OF THE MFID  [0]
     03 FPB.QUEUE.FILE              BOOLEAN,
            % IF 0 THEN SINGLE QUEUE IF 1 THEN MULTIPLE QUEUES  [0]
     03 FPB.DMS.FLAG                BOOLEAN,
            % DMS AUDIT FILE  [0]
```

```
    03 FPB.LDDMP                        BOOLEAN,%
          %THIS WILL BE A LIBRARY TAPE  [0]
    03 FPB.EMULATOR.TAPE                BOOLEAN,
          % NC MCP TAPE SERVICES PROVIDED  [0]
    03 FPB.HEADER                       BOOLEAN,%
          % USERS REMOTE READS AND WRITES WILL INCLUDE THE
          % HEADER  [0]
    03 FPB.TRANSLATE                    BOOLEAN,
          % TRANSLATE ALL RECORDS USING FPB.TRANSLATE.NAME FILE  [0]
    03 FPB.USER.BACKUP.NAME             BOOLEAN,
          % USE USER PRINT FILE NAME FOR NAME OF PRINTER BACKUP FILE
          % RATHER THAN MCP GENERATED NAME.  [0]
    03 FPB.ATTRIBUTE.ERROR              BOOLEAN,
          %INDICATES WHETHER ERROR OCCURRED ON LAST
          % FILE ATTRIBUTE REQUEST  [0]
    03 FPB.DUMMY.FILE                   BOOLEAN,
          % NO LOGICAL I/O TO BE DONE ON THIS FILE  [0]
    03 FPB.COBOL74                      BOOLEAN,
          % ANSI74 COBOL FILE [0]
    03 FPB.SIMPLE.HEADERS               BOOLEAN,%
          % RMT FILE W/ HEADERS BUT NO CONTROL MESSAGES. [0]
    03 FPB.BACKUP.FILE.EXTENTION        BOOLEAN,%
          % EXTENTION FILE WHEN FIRST FILE EXCEEDS FILE SIZE. [0]
    03 FPB.FLEXIBLE                     BIT(1),%
          % IF EFFECT, REQUEST A MAXIMUM OF 105 AREAS.  [0]
 02 FPB.RECORD.SIZE                     WORD,                    %0382
          % MAX LOGICAL RECORD SIZE IN BITS
 02 FPB.RECORDS.PER.BLOCK               WORD,                    %0406
          % NUMBER OF LOGICAL RECORDS PER PHYSICAL RECORD  [1]
 02 FPB.MAX.BLOCK.SIZE                  WORD,                    %0430
          % MAX PHYSICAL RECORD SIZE IN BITS FOR VARIABLE
          % LENGTH RECORDS ONLY. NOT CONSULTED IF
          % FPB.VARIABLE = 0 [FPB.RECORD.SIZE * FPB.RECORDS.PER.
          % BLOCK]
 02 FPB.ADVERB                          BIT (12), %[0]           %0454
    03 FPB.INPUT                        BOOLEAN,
          %
    03 FPB.OUTPUT                       BOOLEAN,
          %
    03 FPB.NEW                          BOOLEAN,
          %
    03 FPB.WITH.PUNCH                   BOOLEAN,
          %
    03 FPB.WITH.PRINT                   BOOLEAN,
          %
    03 FPB.NO.REWIND                    BOOLEAN,
          % DOUBLES WITH "WITH.INTERPRET".
    03 FPB.REVERSE                      BOOLEAN,
          % DOUBLES WITH "WITH STACKERS".
    03 FPB.OPEN.LOCK                    BOOLEAN,
```

```
            %
    03 FPB.OPEN.LOCKOUT                 BOOLEAN,
            %
    03 FPB.REPORT.NO.FILE               BOOLEAN,
            %
    03 FPB.REPORT.LOCKED.FILE           BOOLEAN,
            %
    03 FPB.OPEN.CODE                    BOOLEAN,

            % ADVERB USED FOR IMPLIED OPENS
 02 FILLER                              BIT (80),                    %0466
            % 80 AVAILABLE ONES  [0]
 02 FPB.LABEL.TYPE                      BIT (4),                     %0546
            % 0 = DEFAULT ANSI LABEL  [0 is the default]
            % 1 = UNLABELED
            % 2 = 7 CHAR BURROUGHS STANDARD
 02 FPB.SAVE                            WORD,                        %0550
            % SAVE FACTOR FOR MAG TAPE AND DISK  [1]
 02 FPB.REEL                            WORD,                        %0574
            % REEL NUMBER  [1]
 02 FPB.EXCEPTION.MASK                  WORD,                        %0598
            %  [0]
 02 FPB.UNIT.NAME                       CHAR (6),                    %0622
            % MNEMONIC OF DESIRED UNIT  [blank]
 02 FPB.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]
%
%
%%%%%%%%%%THE FOLLOWING ARE FOR MICR USE%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%
 02 FPB.USE.ROUTINE                     BIT (32),                    %0678
            % SEGMENT AND DISPLACEMENT OF THE FIRST INSTRUCTION
            % IN THE USE ROUTINE  [0]
   10 FPB.DMS.AUDIT.SERIAL.NUMBER       BIT(32),          % DMS
            % SERIAL NUMBER OF FIRST AUDIT IN CURRENT AUDIT FILE
 02 FPB.USE.AREA                        BIT (48),                    %0710
            % A 24 BIT LENGTH AND 24 BIT ADDRESS
            % OF THE USE ROUTINE WORK AREA  [0]
 02 FPB.SR.STATION                      BIT (4),                     %0758
            % READ STATION FOR READER/SORTER  [0]
 02 FPB.ACCESS                          BIT (4),                     %0762
            % 0 = SERIAL  [0 is the default]
            % 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 FPB.AREAS                          WORD,                          %0766
            % MAX AREAS DESIRED
            % LIMIT = 105  [25]
  02 FPB.BLCKS.AREA                     WORD,                          %0790
            % NUMBER OF PHYSICAL RECORDS PER AREA  [100]
  02 FPB.EU.DRIVE                       BIT (4),                       %0814
            % SPECIAL EU OR DRIVE.NO USED WITH
            % FPB.INC.EU AND FPB.SPECIAL.EU  [0]
  02 FPB.ALL.AT.OPEN                    BOOLEAN,                       %0818
            % ALLOCATE ALL AREAS AT OPEN TIME  [0]
  02 FPB.CYL.BOUNDARY                   BOOLEAN,                       %0819
            % ALLOCATE AREAS ON A CYLINDER BOUNDARY  [0]
  02 FPB.MULTI.PACK.FILE                BOOLEAN,                       %0820
            % FILE CAN GO ON MULTI-PACK  [0]
  02 FPB.SPECIAL.EU                     BOOLEAN,                       %0821
            % FILE MUST GO ON EU OR DRIVE SPECIFIED BY FPB.EU.DRIVE
            % [0]
  02 FPB.INC.EU                         BOOLEAN,                       %0822
            % INCREMENT EU OR DRIVE FOR EACH AREA  [0]
  02 FPB.RESTORE.IMAGE                  BIT (1),%                      %0823
            % RESTORE FPB IMAGE ON CLOSE RELEASE LOCK REMOVE ETC.
            % RESERVED FOR MCP USE ONLY.
  02 FPB.EXTENDED.SEQUENTIAL            BOOLEAN,                       %0824
            % USED TO INDICATE EXTENDED.SEQUENTIAL ACCESS WHEN
            %    FPB.ACCESS = 0 OR FPB.ACCESS = 2.  [1]
  02 FPB.LINEFORMAT                     BIT(1)                         %0825
            % IF SET, WE WILL FORMAT PAGE ACCORDING TO PARAMS BELOW.
  FPB.CONTINUE.1#,%
FPB.CONTINUE.1 AS #,%
 %%%%%%%%%%THE FOLLOWING ARE FOR MCP USE ONLY%%%%%%%%%%%%%%%%%%%%%%%%%
 %
 %
  02 FPB.AUTOPRINT                      BIT (1),                       %0826
            % 0=AUTOPRINTABLE          [0]
  02 FPB.SYSTEM.BACKUP.BIT              BIT (1),                       %0827
            % TO IDENTIFY AUTO BACKUP PROGRAM, SYSTEM/BACKUP
            % RESERVED FOR MCP USE ONLY.
  02 FPB.REPETITIONS                    BIT(6),                        %0828
            % NUMBER OF COPIES OF BACKUP FILE  [0]
  02 FPB.PAGE.SIZE                      BIT(8),                        %0834
            %
  02 FPB.UPPER.MARGIN                   BIT(8),                        %0842
            %
  02 FPB.LOWER.MARGIN                   BIT(8),                        %0850
            %
  02 FPB.FOOTING                        BIT(8),                        %0858
            %
  02 FPB.INITIALIZE.RELATIVE.AREAS      BOOLEAN,                       %0866
            %
```

```
  02 FILLER                              BIT(127),                    %0867
            %127 FREE ONES  [0]
  02 FPB.MCPDATA                         OSK.ADR,                     %0994
            % RESERVED FOR MCP USE ONLY.
  02 FPB.MCPINTERNAL                     BOOLEAN,                     %1030
            % RESERVED FOR MCP USE ONLY.
  02 FPB.BACKUP.ALREADY                  BOOLEAN,                     %1031
            % THIS IS A BACKUP FILE - DO NOT PUT TO BACKUP AGAIN
            % RESERVED FOR MCP USE ONLY.
  02 FPB.NEW.FORMAT                      WORD,                        %1032
            %THIS WILL = -1 FOR ONE SEGMENT FPB
            %THIS WILL = -2 FOR TWO SEGMENT FPB
            %THIS WILL = -3 FOR TWO SEGMENT WITH NEW FPB.FILE.TYPE
  02 FPB.OLD.FILE.TYPE                   BIT (4),                     %1056
            % IF FPB.NEW.FORMAT EQL -3 THEN THIS
            % FIELD HAS BEEN REPLACED BY FPB.FILE.TYPE
  02 FPB.PSEUDO.ADR                      ADDRESS,                     %1060
            % PSEUDO READER FOR THIS FILE  [0]
  02 FPB.SAVE.HDWR                       BIT(6),                      %1084
            % SAVE HDWR TYPE IF EVER CHANGED BY OPEN
            % RESERVED FOR MCP USE ONLY.
  02 FPB.INV.CHARS                       BIT(2),%                     %1090
            % 0 = REPORT ALL INVALID CHARACTER LINES
            % 1 = REPORT ALL AND STOP AT THAT LINE
            % 2 = REPORT FIRST ONE ONLY  [default]
            % 3 = DONT REPORT ANY INVALID CHARACTERS
  02 FPB.SERIAL                          CHAR (6),%                   %1092
            %ALPHA-NUMERIC TAPE SERIAL  ID  [blanks]
%
%%%%%%%%%% FOR QUEUE/REMOTE FILES ONLY %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
  02 FPB.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 FPB.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 FPB.TRANSLATE.NAME                 CHARACTER(10)                %1156
            %TRANSLATE FILE NAME ("TRANSLATE"/FPB.TRANSLATE.NAME)
            % [blank]
* 02 FPB.PROTECTION                     BIT (2) %  HOST RJE          %1236
            % 0 = PUBLIC FILE  [default]
            % 1 = PRIVATE FILE
            % 2 = GUARD FILE
* 02 FPB.PROTECTION.IO                  BIT (2)%       HOST RJE      %1238
```

```
                % 0 = I/O  [default]
                % 1 = INPUT ONLY
                % 2 = OUTPUT ONLY
* 02 FPB.MERGED.ATTRIBUTES              BIT(12)                          %1240
                % RESERVED FOR MCP USE ONLY.
  %    MERGED ATTRIBUTES FROM FPB AND OPEN
  %    ASSISTS IL UL AND OU AND DATA.RECORDER IO.UP GENERATION
  %    CREATED BY DR.STANDARDIZE.OPEN IN OPENERS
  %    FOR DATA.RECORDER.FILES ONLY AT THIS TIME]]]   ]]]
*     05 FILLER                        BIT(1)          % 0
*     05 FPB.MERGED.INPUTF             BOOLEAN         % 1
*     05 FPB.MERGED.OUTPUTF            BOOLEAN         % 2
*     05 FPB.MERGED.OPEN.OPTIONS       BIT(4)          % 3 TO 6
*        10 FPB.MERGED.WITH.PUNCH      BOOLEAN         % 3
*        10 FPB.MERGED.WITH.PRINT      BOOLEAN         % 4
*        10 FPB.MERGED.WITH.INTERPRET  BOOLEAN         % 5
*        10 FPB.MERGED.WITH.STACKERS   BOOLEAN         % 6 DANGER
*           11 FPB.MERGED.REVERSEF     BOOLEAN         % 6 DANGER
*     05 FPB.MERGED.LOCKF              BOOLEAN         % 7
*     05 FPB.MERGED.LOCKOUTF           BOOLEAN         % 8
*     05 FPB.MERGED.RRRNOFILE          BOOLEAN         %  9
*     05 FPB.MERGED.RRRLOCK.FILE       BOOLEAN         % 10
*     05 FPB.MERGED.OPENCODE           BOOLEAN         % 11
* 02 FPB.MLTI.CHGD                     BIT (2)%                         %1252
                % RESERVED FOR MCP USE ONLY.
* 02 FPB.PKID.CHGD                     BIT (1)%                         %1254
                % RESERVED FOR MCP USE ONLY.
* 02 FPB.PROTOCOL                      BIT (8)%                         %1255
                % USED BY APPLICATION PROGRAM TO TELL MCS MESSAGE FORMAT
                % [0]
* 02 FPB.EXPANDED.ADVERB               BIT(12)  % [0]                   %1263
*     03 FPB.INTERPRET                 BOOLEAN%
                %
*     03 FPB.STACKERS                  BOOLEAN%
                % NO DOUBLE USES IN EXPANDED OPEN;
*     03 FPB.IN.SECURE                 BOOLEAN%
                % OVERRIDE RJE NAMING CONVENTIONS AND SECURITY;
*     03 FPB.OPEN.ON.BEHALF.OF         BOOLEAN%
                % DO NOT USE PROG'S USERCODE/PASSWORD;
*     03 FILLER                        BIT(8)%
                % 8 BITS AVAILABLE;
                % EXPANDED ADVERB USED ONLY BY EXPANDED OPEN;
* 02 FPB.OBO.UC.INFO                BIT(12)%                            %1275
*     03 FPB.OBO.UC.USED            BOOLEAN%
                % OBO OPEN BIT WAS USED - REMEMBER FOR FILE CLOSE.
*     03 FPB.OBO.UC.PRIV            BOOLEAN%
                % FILE OPENED ON BEHALF OF PRIV USERCODE.
*     03 FPB.OBO.UC.INDEX           BIT(10)%
                % USERCODE INDEX FOR WHICH FILE WAS OPENED ON BEHALF OF
* 02 FILLER                            BIT (153)                        %1287
```

```
            % AVAILABLE SPACE FOR GLOBAL FILE ATTRIBUTES  [0]
, 02 FPB.NUMBER.STATIONS.NEW          BIT(12)                    %1440
            % IF FPB.NEW.FORMAT = -2
            % THEN NUMBER OF REMOTE STATIONS ASSIGNED TO THIS FILE [0]
, 02 FPB.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 FPB.IS.SUB.FPB.PTR               BIT(12)                    %1464
            % INDEXED/SEQUENTIAL - SUB FILE FPB OFFSET IN SEGMENTS
            %                      FROM BEGINNING OF DISK CODE FILE.
            %                      [0]
, 02 FPB.IS.NUMBER.SUB.FPBS           BIT(8)                     %1476
            % INDEXED/SEQUENTIAL - NUMBER OF KEYS + (1 FOR DATA FILE).
            %                      [0]
, 02 FILLER                           BIT(12)                    %1484
, 02 FPB.IS.NUM.IO.DESC               BIT(6) % [0]               %1496
, 02 FPB.AREALENGTH                   BIT(24)                    %1502
            % USED IN CONJUNCTION WITH:  FPB.RECORD.SIZE
            %
            %                           FPB.BLOCK.SIZE
            %
            %
            %                           FPB.FRAMESIZE
, 02 FPB.BLOCK.SIZE                   BIT(24)                    %1526
            % USED IN CONJUNCTION WITH:  FPB.RECORD.SIZE
            %
            %
            %                           FPB.FRAMESIZE
, 02 FPB.DENSITY                      BIT(24)                    %1550
            % 1 = BPI200 , 2 = BPI556 , 3 = BPI800 , 4 = BPI1600  [4]
, 02 FPB.FRAMESIZE                    BIT(24)                    %1574
            % THE NUMBER OF BITS TO BE TRANSFERRED AS A UNIT OF DATA.
            % [DEFAULT ON B1700/B1800 SHOULD BE 1.]
, 02 FPB.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 FPB.IS.KEY.PRIME                 BOOLEAN                    %1610
            % INDEXED/SEQUENTIAL - THIS FILE CONTAINS THE PRIME KEYS.
, 02 FPB.IS.KEY.DUP.ALLOWED           BOOLEAN                    %1611
            % INDEXED/SEQUENTIAL - MAY THIS FILE CONTAIN DUPLICATE
            %                      KEYS.
, 02 FPB.IS.KEY.OFFSET                BIT(16)                    %1612
            % INDEXED/SEQUENTIAL - BIT OFFSET IN DATA RECORD TO
            %                      BEGINNING OF THIS KEY.
, 02 FPB.IS.KEY.SIZE                  BIT(12)                    %1628
            % INDEXED/SEQUENTIAL - BIT SIZE OF THIS KEY.
, 02 FPB.IS.KEY.SIGNED                BOOLEAN                    %1640
            % INDEXED/SEQUENTIAL - THIS FILE CONTAINS SIGNED KEYS.
, 02 FPB.IS.KEY.DECENDING             BOOLEAN                    %1641
            % INDEXED/SEQUENTIAL - THESE KEYS ARE IN DECENDING ORDER.
```

```
, 02 FILLER                                BIT(2)                          %1642
, 02 FPB.MINRECSIZE                        BIT(24)                         %1644
            % MINIMUM RECORD SIZE FOR VARIABLE LENGTH RECORDS
, 02 FPB.HOSTNAME                          CHAR(17)
            % BNA/HOST SERVICES.
;
#;%
 %
 %
 %
```

IPB DECLARATIONS

```
DEFINE IPB.SIZE AS #1440#;
 %
    DEFINE IPB.DECLARATION AS#
    DECLARE 01 DUMMY REMAPS IPB BIT(1440),
               02 FILLER BIT(1192),
               02 IPB.HARDWARE CHAR(1),
               02 IPB.ARCHITECTURE.NAME CHAR(10),
               02 IPB.COMPILER.LEVEL BIT(8),
               02 IPB.MCP.LEVEL BIT(8),
               02 IPB.GISMO.LEVEL BIT(8),
               02 IPB.ARCHITECTURE.ATTRIBUTES BIT(80),
               02 IPB.DATA.SPACE.SIZE BIT(24),
               02 IPB.PRE.INIT.DATA.SIZE BIT(16),
               02 IPB.DATA.PTR BIT(16)
#;
 %
 %
 %
 %
 %
 %
```

HINTS

```
 % THE FOLLOWING DECLARATIONS MUST ALWAYS REMAIN IN THE BEGINNING AND
 % THEIR ORDER MUST NOT BE CHANGED. THE FIRST THREE ARE RESERVED FOR THE
 % SDL INTERPRETER, THE REST ARE FOR THE DUMP ANALYZER.
 %
DECLARE HINTS BIT(2088);
 %
DEFINE%
 %   DISPATCH.WORD            AS #SUBBIT(HINTS,0000,024)#%  HEX  000
 %   ADDR.DP.PROC2            AS #SUBBIT(HINTS,0024,024)#%  HEX  018
     DISK.TRACE.BLOCK.ADDR    AS #SUBBIT(HINTS,0048,024)#%  HEX  030
 %   MASTER.GISMO             AS #SUBBIT(HINTS,0072,024)#%  HEX  048
```

```
,   LOCN.MAKE.MCP.BE.HERE     AS #SUBBIT(HINTS,0096,033)#%  HEX  060
,   LOCN.INTERP.DICT          AS #SUBBIT(HINTS,0129,024)#%  HEX  081
,   KI.KO                     AS #SUBBIT(HINTS,0153,001)#%  HEX  099
,   NO.REINSTATES             AS #SUBBIT(HINTS,0154,001)#%  HEX  09A
,   FIRE.UP.CONTROLLER        AS #SUBBIT(HINTS,0155,001)#%  HEX  09B
,   N.SECOND.COUNTER          AS #SUBBIT(HINTS,0156,002)#%  HEX  09C
,   MCP.LIMIT                 AS #SUBBIT(HINTS,0158,024)#%  HEX  09E
,   HINTS.LAST.OVLY           AS #SUBBIT(HINTS,0182,024)#%  HEX  0B6
,   MICR.DEBUG.BIT              AS #SUBBIT(HINTS,0206,1)#%  HEX 0CE
,   C.NOT.LOCKED              AS #SUBBIT(HINTS,0207,001)#%  HEX  0CF
,   DFH.DIR.AD                AS #SUBBIT(HINTS,0208,024)#%  HEX  0D0
,   AUTO.GUARD                AS #SUBBIT(HINTS,0232,003)#%  HEX  0E8
,   FIRE.SYSTEM.BACKUP        AS #SUBBIT(HINTS,0235,003)#%  HEX  0EB
,   FOUND.BACKUP.DESIGNATION  AS #SUBBIT(HINTS,0238,001)#%  HEX  0EE
%   INTERRUPT.DISABLE.BIT     AS #SUBBIT(HINTS,0239,001)#%  HEX  0EF
%   TRACE.SWITCHES            AS #SUBBIT(HINTS,0240,027)#%  HEX  0F0
,   TRACE.WORD                AS #SUBBIT(HINTS,0243,024)#%  HEX  0F3
,   TRACE.FLAG                AS #SUBBIT(HINTS,0261,003)#%  HEX  105
,   TRACE.OPT                 AS #SUBBIT(HINTS,0264,003)#%  HEX  108
,   HINTS.FIRST.QUEUE         AS #SUBBIT(HINTS,0267,024)#%  HEX  10B
,   ADDR.OF.COLD.START.VAR    AS #SUBBIT(HINTS,0291,024)#%  HEX  123
%   ADDR.OF.INTERRUPT.INFO    AS #SUBBIT(HINTS,0315,024)#%  HEX  13B
,   MAXM                      AS #SUBBIT(HINTS,0339,004)#%  HEX  153
,   ELOG.FULL                 AS #SUBBIT(HINTS,0343,001)#%  HEX  157
,   NON.RELEASE.MCP           AS #SUBBIT(HINTS,0344,001)#%  HEX  158
,   GISMO.LEVEL               AS #SUBBIT(HINTS,0345,008)#%  HEX  159
%   HI.RESOLUTION.TIMER.SET   AS #SUBBIT(HINTS,0353,001)#%  HEX  161
,   PSR.CHANGE.BIT            AS #SUBBIT(HINTS,0354,001)#%  HEX  162
,   RELEASE.LEVEL             AS #SUBBIT(HINTS,0355,008)#%  HEX  163
,   FIRST.LINK                AS #SUBBIT(HINTS,0363,024)#%  HEX  16B
%   MCP.TYPE                  AS #SUBBIT(HINTS,0387,004)#%  HEX  183
,   SYCOUNTER                 AS #SUBBIT(HINTS,0391,020)#%  HEX  187
,   SY.CNTR.MSK               AS #SUBBIT(HINTS,0411,020)#%  HEX  19B
,   SY.PRIOR.TIME             AS #SUBBIT(HINTS,0431,020)#%  HEX  1AF
%   RESERVED.FOR.INTERP.USE   AS #SUBBIT(HINTS,0451,020)#%  HEX  1C3
,   SYSTEM.PACK.INFO          AS #SUBBIT(HINTS,0471,024)#%  HEX  1D7
%   TRACE.PORT.CHANNEL        AS #SUBBIT(HINTS,0495,007)#%  HEX  1EF
,   SYSTEM.UNIT               AS #SUBBIT(HINTS,0502,012)#%  HEX  1F6
,   SYSTEM.PORT.CHAN          AS #SUBBIT(HINTS,0502,007)#%  HEX  1F6
,   SYSTEM.PORT               AS #SUBBIT(HINTS,0502,003)#%  HEX  1F6
,   SYSTEM.CHANNEL            AS #SUBBIT(HINTS,0505,004)#%  HEX  1F9
%   DUMMY.BIT.RESERVED        AS #SUBBIT(HINTS,0509,001)#%  HEX  1FD
,   SYSTEM.UNIT.EU            AS #SUBBIT(HINTS,0510,004)#%  HEX  1FE
,   CONSOUL.SWITCHES          AS #SUBBIT(HINTS,0514,028)#%  HEX  202
,   MICRO.TRACE.FLAG          AS #SUBBIT(HINTS,0542,001)#%  HEX  21E
%   GISMO.TRACE.SPACE         AS #SUBBIT(HINTS,0543,024)#%  HEX  21F
,   PORT.CHANNEL.TABLE        AS #SUBBIT(HINTS,0567,192)#%  HEX  237
,   BYPASS.CLEANUP            AS #SUBBIT(HINTS,0759,001)#%  HEX  2F7
,   CONTRL.CRD.FLG            AS #SUBBIT(HINTS,0760,001)#%  HEX  2F8
,   EXT.RESULT.DESC.CHAIN     AS #SUBBIT(HINTS,0761,024)#%  HEX  2F9
```

```
,    T.FILES                      AS #SUBBIT(HINTS,0785,008)#%  HEX  311
,    MICR.COUNT                   AS #SUBBIT(HINTS,0793,006)#%  HEX  319
,    CHANGE.BIT                   AS #SUBBIT(HINTS,0799,001)#%  HEX  31F
,    RELEASE.VERSION              AS #SUBBIT(HINTS,0800,008)#%  HEX  320
%    FILLER                       AS #SUBBIT(HINTS,0808,001)#%  HEX 328
,    IOAT.POINTER                 AS #SUBBIT(HINTS,08C9,024)#%  HEX  329
,    IOAT.END                     AS #SUBBIT(HINTS,0833,024)#%  HEX  341
,    SYSTEM.PAUSE.DESC            AS #SUBBIT(HINTS,0857,024)#%  HEX  359
,    PSEUDO.TABLE.ADDRESS         AS #SUBBIT(HINTS,0881,024)#%  HEX  371
,    DATE.SET                     AS #SUBBIT(HINTS,0905,001)#%  HEX  389
,    TIME.SET                     AS #SUBBIT(HINTS,0906,001)#%  HEX  38A
,    GISMO.OPTIONS                AS #SUBBIT(HINTS,0907,024)#%  HEX  38B
,    CHECK.RA                     AS #SUBBIT(HINTS,0907,001)#%  HEX  38B
,    COMM.TRACE                   AS #SUBBIT(HINTS,0908,001)#%  HEX  38C
,    GISMO.TRACE                  AS #SUBBIT(HINTS,0909,001)#%  HEX  38D
,    XFER.24                      AS #SUBBIT(HINTS,0910,001)#%  HEX  38E
,    CMS                          AS #SUBBIT(HINTS,0911,001)#%  HEX  38F
,    READR.SORTR                  AS #SUBBIT(HINTS,0912,001)#%  HEX  390
,    ANY.MAG.TAPE                 AS #SUBBIT(HINTS,0913,001)#%  HEX  391
,    NRZ.MAG.TAPE                 AS #SUBBIT(HINTS,0914,001)#%  HEX  392
,    CASSETT                      AS #SUBBIT(HINTS,0915,001)#%  HEX  393
,    PAPER.TAPE                   AS #SUBBIT(HINTS,0916,001)#%  HEX  394
,    DATA.COMM                    AS #SUBBIT(HINTS,0917,001)#%  HEX  395
,    PORT.DEVICES                 AS #SUBBIT(HINTS,0918,001)#%  HEX  396
,    EXCHANGES                    AS #SUBBIT(HINTS,0919,001)#%  HEX  397
,    PORT.TRACE                   AS #SUBBIT(HINTS,0920,001)#%  HEX  398
,    B1720.CODE                   AS #SUBBIT(HINTS,921,001)#%   HEX  399
,    B1860.CODE                   AS #SUBBIT(HINTS,0922,001)#%  HEX  39A
,    MPROC.CODE                   AS #SUBBIT(HINTS,0923,001)#%  HEX  39B
,    B1830.CODE                   AS #SUBBIT(HINTS,0924,001)#%  HEX  39C
,    READ.AFTER.WRITE.CHECK       AS #SUBBIT(HINTS,0925,001)#%  HEX  39D
,    PRIORITY.MEMORY.MGMT         AS #SUBBIT(HINTS,0926,001)#%  HEX  39E
,    FIFO.MEMORY.MGMT             AS #SUBBIT(HINTS,0927,001)#%  HEX  39F
,    THRASHING.COUNTING           AS #SUBBIT(HINTS,0928,001)#%  HEX  3A0
%    RESERVED FOR SYSTEM/INIT     AS #SUBBIT(HINTS,0929,002)#%  HEX  3A1
,    CCH.SCRATCH.MEM.ADDR         AS #SUBBIT(HINTS,0931,024)#%  HEX  3A3
,    S.MCP.TRACE                  AS #SUBBIT(HINTS,0955,001)#%  HEX  3BB
,    INTERRUPT.SWITCH.SET         AS #SUBBIT(HINTS,0956,001)#%  HEX  3BC
,    CISAELE.INTERRUPT.SW         AS #SUBBIT(HINTS,0957,001)#%  HEX  3BD
,    MEMORY.USAGE.BIT             AS #SUBBIT(HINTS,0958,001)#%  HEX  3BE
,    MIN.MEMORY.SIZE              AS #SUBBIT(HINTS,0959,012)#%  HEX  3BF
,    BEEN.THRU.MCP.BE.HERE        AS #SUBBIT(HINTS,0971,001)#%  HEX  3CB
,    REMOTE.REROUTE               AS #SUBBIT(HINTS,0972,001)#%  HEX  3CC
,    QUEUE.REROUTE                AS #SUBBIT(HINTS,0973,001)#%  HEX  3CD
%    DISK.MONITOR.GISMO           AS #SUBBIT(HINTS,0974,004)#%  HEX  3CE
,    SPOLOG.NEEDS.TRANSFERING     AS #SUBBIT(HINTS,0978,001)#%  HEX  3D2
,    INTERPRETER.TABLE.ADDR       AS #SUBBIT(HINTS,0979,036)#%  HEX  3D3
,    SPO.PORT.CHAN                AS #SUBBIT(HINTS,1015,007)#%  HEX  3F7
,    SPO.PORT                     AS #SUBBIT(HINTS,1015,003)#%  HEX  3F7
,    SPO.CHANNEL                  AS #SUBBIT(HINTS,1018,004)#%  HEX  3FA
```

```
,    KEYBOARD.SPO.DESC                AS #SUBBIT(HINTS,1022,024)#%  HEX  3FE
,    SPO.SQ.AD                        AS #SUBBIT(HINTS,1046,24)#%   HEX  416
%    CHANNELS.NOT.PRESENT             AS #SUBBIT(HINTS,1070,016)#%  HEX  42E
,    LAMP.FLAGS                       AS #SUBBIT(HINTS,1086,024)#%  HEX  43E
,    LAMP.DISK.TABLE.ADDR             AS #SUBBIT(HINTS,1110,024)#%  HEX  456
,     VARIABLE.LAMP.ALL.USER.CPU            AS#SUBBIT(HINTS,1113,1)#% 459
,     VARIABLE.LAMP.ALL.USER.CODE.OVLY AS#SUBBIT(HINTS,1114,1)#% 45A
,     VARIABLE.LAMP.ALL.USER.DATA.OVLY AS#SUBBIT(HINTS,1115,1)#% 45B
,     VARIABLE.LAMP.SMCP.CPU.1ST.BIT   AS#SUBBIT(HINTS,1122,1)#% 462
,     VARIABLE.LAMP.SMCP.OVLY          AS#SUBBIT(HINTS,1128,2)#% 468
,    SEGMENT.HALT                     AS #SUBBIT(HINTS,1134,004)#%  HEX  46E
,    HALT.MASK                        AS #SUBBIT(HINTS,1138,024)#%  HEX  472
,    MMCP.SEGMENT.HALT                AS #SUBBIT(HINTS,1162,004)#%  HEX  48A
,    MMCP.HALT.MASK                   AS #SUBBIT(HINTS,1166,024)#%  HEX  48E
,    COMPILE.TIME.OPTIONS             AS #SUBBIT(HINTS,1190,008)#%  HEX  4A6
,    RELEASE.VERSION.MCP              AS #SUBBIT(HINTS,1190,001)#%  HEX  4A6
,    DEBUG.OPTION                     AS #SUBBIT(HINTS,1191,001)#%  HEX  4A7
%    6 MORE OPTIONS                   AS #SUBBIT(HINTS,1192,006)#%  HEX  4A8
,    CONTROL.MEMORY.SIZE              AS #SUBBIT(HINTS,1198,004)#%  HEX  4AE
,    MCP.VERSION.DATE                 AS #SUBBIT(HINTS,1202,016)#%  HEX  4B2
,    MAIN.MEMORY.SIZE                 AS #SUBBIT(HINTS,1218,012)#%  HEX  4C2
,    DM.GLOBALS                       AS #SUBBIT(HINTS,1230,024)#%  HEX  4CE
,    QUEUE.ROOT.ADDRESS               AS #SUBBIT(HINTS,1254,024)#%  HEX  4E6
%    MMCP.DATA.PTR                    AS #SUBBIT(HINTS,1278,024)#%  HEX  4FE
,    DC.CHAIN                         AS #SUBBIT(HINTS,1302,024)#%  HEX  516
,    MESSAGE.LIST.TAIL                AS #SUBBIT(HINTS,1326,024)#%  HEX  52E
,    ABSOLUTE.S.MEM                   AS #SUBBIT(HINTS,1350,012)#%  HEX  546
,    S.C.Q.EV                         AS #SUBBIT(HINTS,1362,001)#%  HEX  552
,    M.C.Q.EV                         AS #SUBBIT(HINTS,1363,001)#%  HEX  553
,    S.M.Q.EV                         AS #SUBBIT(HINTS,1364,001)#%  HEX  554
,    S.I.Q.EV                         AS #SUBBIT(HINTS,1365,001)#%  HEX  555
,    M.M.Q.EV                         AS #SUBBIT(HINTS,1366,001)#%  HEX  556
,    M.I.Q.EV                         AS #SUBBIT(HINTS,1367,001)#%  HEX  557
,    M.CAUSE.LOCK                     AS #SUBBIT(HINTS,1368,001)#%  HEX  558
%    M.EV.FILLER                      AS #SUBBIT(HINTS,1369,005)#%  HEX  559
,    M.MCP.LR                         AS #SUBBIT(HINTS,1374,024)#%  HEX  55E
%    M.MCP.OTHERS                     AS #SUBBIT(HINTS,1398,048)#%  HEX  576
%    M.LRU.SPACE                      AS #SUBBIT(HINTS,1446,096)#%  HEX  5A6
,    M.NUMBER.PAGES                   AS #SUBBIT(HINTS,1542,024)#%  HEX  606
,    TRACE.ADDR                       AS #SUBBIT(HINTS,1566,024)#%  HEX  61E
,    M.MCP.Q.IDENT                    AS #SUBBIT(HINTS,1590,008)#%  HEX  636
,    COMM.SPLITTER.ADDR               AS #SUBBIT(HINTS,1598,024)#%  HEX  63E
,    COMM.SPLITTER.LENGTH             AS #SUBBIT(HINTS,1622,016)#%  HEX  656
,    FIRST.RUN.UNIT                   AS #SUBBIT(HINTS,1638,024)#%  HEX  666
,    INDEX.SEQ.USER.COUNT             AS #SUBBIT(HINTS,1662,008)#%  HEX  67E
,    MIKES.HALT.SPACE                 AS #SUBBIT(HINTS,1670,096)#%  HEX  686
%    GISMO.OPTIONS.TWO                AS #SUBBIT(HINTS,1766,024)#%  HEX  6E6
,    LAMP.CPU.BASE                    AS #SUBBIT(HINTS,1766,001)#%  HEX  6E6
,    FIXED.LAMP.DISPLAY               AS #SUBBIT(HINTS,1767,001)#%  HEX  6E7
,    VAR.LAMP.BASE                    AS #SUBBIT(HINTS,1768,001)#%  HEX  6E8
```

```
,   VAR.LAMP.CPU.OVLY          AS #SUBBITCHINTS,1769,001)#%  HEX  6E9
,   VAR.LAMP.IO                AS #SUBBITCHINTS,1770,001)#%  HEX  6EA
,   VAR.LAMP.BARGRAPH          AS #SUBBITCHINTS,1771,001)#%  HEX  6EB
,   MEMORY.BASE.UNIT.5         AS #SUBBITCHINTS,1772,001)#%  HEX  6EC
%   17 MORE CISMO OPTIONS
,   LAST.MEM.LINK              AS #SUBBITCHINTS,1790,024)#%  HEX  6FE
,   SMCP.CPU.PRIORITY          AS #SUBBITCHINTS,1814,024)#%  HEX  716
%   14 FREE ONES               AS #SUBBITCHINTS,1838,014)#%  HEX  72E
,   JOBS.SWEEPS.BEFORE.DECAY   AS #SUBBITCHINTS,1852,010)#%  HEX  73C
,   SYSTEM.IC                  AS #SUBBITCHINTS,1862,012)#%  HEX  746
,   CPU.ID                     AS #SUBBITCHINTS,1862,004)#%  HEX  746
%     0=ERROR      1=B1710      2=B1720
%                  3=B1830      4=B1860
,   MEMORY.IC                  AS #SUBBITCHINTS,1866,004)#%  HEX  74A
%     0=DEFAULT    1=CORRECTABLE S-MEMORY PARITY
,   IO.ID                      AS #SUBBITCHINTS,1870,004)#%  HEX  74E
%     0=DEFAULT

,   ELOG.HERE                  AS #SUBBITCHINTS,1874,024)#%  HEX  752
,   QLOCK.COUNT                AS #SUBBITCHINTS,1898,004)#%  HEX  76A
,   CHIP.TABLE.ADDRESS         AS#SUB3ITC HINTS,1902,024)#%  HEX 76E
,   MIX.MEMORY.PRIORITIES      AS #SUBBITCHINTS,1926,016)#%  HEX  786
,   STOP.SCHED.INPUT           AS #SUBBITCHINTS,1942,001)#%  HEX  796
,   NSEC.DISABL.THRASH.FAULT   AS #SUBBITCHINTS,1943,001)#%  HEX  797
,   DISABLE.THRASHING.FAULT    AS #SUBBITCHINTS,1944,001)#%  HEX  798
,   MCP.VARIABLE.MEM.PRIORITY AS #SUBBITCHINTS,1945,005)#%  HEX  799
%   FILLER                     AS #SUBBITCHINTS,1950,002)#%  HEX  79E
,   MEM.SWEEP.PENDING          AS #SUBBITCHINTS,1952,001)#%  HEX  7A0
,   SAMPLING.CLOCK             AS #SUBBITCHINTS,1953,006)#%  HEX  7A1
,   SAMPLING.INTERVAL          AS #SUBBITCHINTS,1959,006)#%  HEX  7A7
,   MEM.SWEEP.INTERVAL         AS #SUBBITCHINTS,1965,010)#%  HEX  7AD
,   MAX.SWEEP.INTERVAL         AS #SUBBITCHINTS,1975,010)#%  HEX  7B7
,   MEM.EXTEND.COUNT           AS #SUBBITCHINTS,1985,002)#%  HEX  7C1
,   OVERLAY.COUNTER            AS #SUBBITCHINTS,1987,008)#%  HEX  7C3
,   OVERLAY.TARGET             AS #SUBBITCHINTS,1995,008)#%  HEX  7CB
,   MCP.SWEEPS.BEFORE.DECAY    AS #SUBBITCHINTS,2003,010)#%  HEX  7D3
,   MEM.DUMP.COMPLETE          AS #SUBBITCHINTS,2013,001)#%  HEX  7DD
,   MAX.MEM.PRIORITY.IN.MIX    AS #SUBBITCHINTS,2014,005)#%  HEX  7DE
,   CONTROLLER.SCHEDULED       AS #SUBBITCHINTS,2019,001)#%  HEX  7E3
%   2 FREE ONES                AS #SUBBITCHINTS,2020,002)#%  HEX  7E4
,   FOUNTAIN                   AS #SUBBITCHINTS,2022,024)#%  HEX  7E6
,   CLEAR.START.REQD           AS #SUBBITCHINTS,2046,001)#%  HEX  7FE
,   SCHEDULER.BLOCK.COUNT      AS #SUBBITCHINTS,2047,008)#%  HEX  7FF
%   1 AVAILABLE BIT            AS #SUBBITCHINTS,2055,001)#%  HEX  807
,   DCPU.DATA                  AS #SUBBITCHINTS,2056,024)#%  HEX  808
,   SLAVE.PRESENT              AS #SUBBITCHINTS,2080,001)#%  HEX  820
,   SLAVE.PORT.CHANNEL         AS #SUBBITCHINTS,2081,007)#%  HEX  821
;
%
%
```

%
%

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.)

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 (FP8.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 (FP8.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 (20) is used when creating a new disk file. The maximum value is 105.

ATTERR ---5 (FP8.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.

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 (FP3.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 (FP8.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 FP8.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 = DONTBACKUP - 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.

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.

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 labeled 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.

BCL 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 invloved 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.

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). The attribute may only be set while the logical file is closed and not assigned to a physical file. 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 ---1026 (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.

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.

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 (FPB.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 (FIB.KEY)

DISK/TAPE, READ ONLY, OPENED, INTEGER

The attribute NEXTRECORD returns the current position of the file. The SPB 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.

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.

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.

TITLE ---80 (FPB.NAMES)

GENERAL, READ/WRITE, ANYTIME/ANYTIME, 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.

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" cp 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.

473 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 existence 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 out most of this overhead (specifically, waits for trace buffers are masked out). Any traces using the timer should be designed with a skeptical eye to the overhead.

INDEX