

SYMBOL/COPY

Data Documents, Inc.

2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57

```

BEGIN
% COPY/CANDE SOURCE PROGRAM, REVISED 5-72 (SHM)
COMMENT: * TITLE: B5500/B5700 MARK XIV SYSTEM RELEASE
* FILE ID: SYMBOL/COPY TAPE ID: SYMBOL2/FILE000
* THIS MATERIAL IS PROPRIETARY TO BURROUGHS CORPORATION
* AND IS NOT TO BE REPRODUCED, USED, OR DISCLOSED
* EXCEPT IN ACCORDANCE WITH PROGRAM LICENSE OR UPON
* WRITTEN AUTHORIZATION OF THE PATENT DIVISION OF
* BURROUGHS CORPORATION, DETROIT, MICHIGAN 48232
* COPYRIGHT (C) 1971, 1972 BURROUGHS CORPORATION
* AA320206 AA386657

REAL COMMON;
SAVE ARRAY A[0:30], BUFFER1, BUFFER2[0:300];
SAVE ARRAY N[0:30], PARAMS[0:10], DISKROWS[0:19];
REAL APTR, ASTART, BASE, ENDRESEQ, EOFPTR, ERRSEQ, FID,
FILETYPE, HDRO, HRANGE, I, INCREMENT, J, LINE,
LRANGE, LREC, MFID, NCT, NPARAMS, NPTR, OUTPUTNAME, QUISEQ,
PC, PREVSEQ, PSIZE, RESULT1, RESULT2, RPB, RSIZE, SEQLUC,
SPB, SPR, SRCESEQ, STARTRESEQ, USER, WPB;
BOOLEAN BRAAK, CONCISE, DATAFILE, ERROR, FORMT, MAKETAB, OK,
PRINTER, PUNCH, RESEQ, SEQCHECK, SFLG, WORKFILE;
LABEL ERRER, EXIT;
DEFINE EOFMARK=100000000#, MASK=3"2000000000"#;
DEFINE MSGSIZE=((APTR.[33:15]-ASTART)*8+APTR.[30:3])#;
FILE OUT NEWTAB DISK SERIAL [20:30 ] (2,30,300,SAVE 1);
FILE OUT PRNT
FILE OUT PNCH
%*****
PROCEDURE DISKWAIT(I,S,D);
VALUE I,S,D; REAL I,S,D; ARRAY AL*; COMMUNICATE(-8);
%*****
PROCEDURE GETHEADER(AREA,NAM1,NAM2,NAM3); VALUE NAM1,NAM2,NAM3;
REAL NAM1,NAM2,NAM3; ARRAY AREA[*]; COMMUNICATE("21");
%*****
STREAM PROCEDURE OUTCONV(A,N,SUPRES); VALUE N,SUPRES;
%*****
BEGIN
SI:=LOC N; DI:=A; DS:=8DEC;
SUPRES(DI:=DI-8; DS:=7FILL);
END STREAM PROCEDURE OUTCONV;
%*****
STREAM PROCEDURE MOVE(N,A,B); VALUE N;
BEGIN SI:=A; DI:=B; DS:=N WDS; END;
%*****
REAL STREAM PROCEDURE ADDRESS(A);
BEGIN SI:=A; ADDRESS:=SI; END;
%*****
REAL STREAM PROCEDURE INPCONV(A);
BEGIN SI:=A; DI:=LOC INPCONV; DS:=8OCT; END;
%*****
STREAM PROCEDURE PLACETYPE(FILETYPE,FIIL); VALUE FILETYPE;
%*****
BEGIN
SI:=FIIL; SI:=SI-24; DI:=LOC FIIL; DS:=WDS;
SI:=FIIL; 2(SI:=SI+56); DI:=LOC FIIL; DS:=WDS;
DI:=FIIL; DI:=DI+38; SI:=LOC FILETYPE; SI:=SI+7; DS:=CHR;
END STREAM PROCEDURE PLACETYPE;
%*****
PROCEDURE INXOUT(A,N,T); VALUE N,T; REAL A,N,T;

```

```

00010000
00010100
00010110
00010111
00010112
00010113
00010114
00010115
00010116
00010117
00010118
00010119
00010200
00010300
00010400
00010500
00010600
00010700
00010800
00010900
00011000
00011100
00011200
00011300
00011400
00011500
00011600
00011700
00011800
00011900
00012000
00012100
00012200
00012300
00012400
00012500
00012600
00012700
00012800
00012900
00013000
00013100
00013200
00013300
00013400
00013500
00013600
00013700
00013800
00013900
00014000
00014100
00014200
00014300
00014400
00014500
00014600
00014700
00014800
00014900

```

FILE OUT LPAPRT 3(2,17), 2 UNIT = 5
LPBPRNT 3(2,17); 2 UNIT = 6
SWITCH FILE ~~LPBPRNT~~ 0 1 2
printfiles = PRNT, LPAPRT, LPBPRNT

Data Document Inc.

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57

```

%*****00015000
BEGIN                                00015100
COMMUNICATE(-11);                    00015200
BRAAK:=BOOLEAN(T); % MCP RETURNS 1 IF BREAK OCCURRED 00015300
APTR:=ASTART;                         00015400
END;                                   00015500
%*****00015600
STREAM PROCEDURE PLACESTRING(APTR,STRING,CONVERT); VALUE STRING,CONVERT;00015700
%*****00015800
BEGIN LOCAL SV; LABEL EXIT;          00015900
SI:=LOC STRING; DI:=APTR; DI:=DI+5; DI:=DC; SV:=DI; 00016000
CONVERT(DS:=8DEC; DI:=DI-8; DS:=7FILL; 00016100
SI:=SV; DI:=SV; 8(IF SC=" " THEN SI:=SI+1 ELSE DS:=CHR); 00016200
JUMP OUT TO EXIT);                  00016300
SI:=SI+1; DS:=7CHR;                 00016400
EXIT:                                 00016500
DS:=LIT" "; SV:=DI; SI:=LOC SV; DI:=APTR; DS:=WDS; 00016600
END STREAM PROCEDURE PLACESTRING;   00016700
%*****00016800
STREAM PROCEDURE MESSAGE(APTR,TYPE); VALUE TYPE; 00016900
%*****00017000
BEGIN LOCAL SV; LABEL L0,L1,L2,L3,L4,L5,EXIT; 00017100
DI:=APTR; DI:=DI+5; DI:=DC;         00017200
CI:=CI+TYPE;                         00017300
GO L0; GO L1; GO L2; GO L3; GO L4; GO L5; 00017400
L0: % TYPE=0, ERROR FOR NON-STANDARD RECORD SIZE 00017500
DS:=37LIT"ERROR: FILE RECORDS ARE NON-STANDARD."; 00017600
GO EXIT;                              00017700
L1: % TYPE=1, NUMBER OF RECORDS COPIED 00017800
DS:=34LIT"RECORDS COPIED, LAST RECOR D COPIED="; 00017900
GO EXIT;                              00018000
L2: % TYPE=2, SEQUENCE ERROR 00018100
DS:=16LIT"SEQUENCE ERROR: "; 00018200
GO EXIT;                              00018300
L3: % TYPE=3, HEADING FOR PRINTER OUTPUT 00018400
DS:=40LIT" "; 00018500
L4: % TYPE=4, HEADING FOR CARD PUNCH OUTPUT 00018600
DS:=6LIT"FILE: "; 00018700
GO EXIT;                              00018800
L5: % TYPE=5, SYSTEM ERROR 00018900
DS:=30LIT"SYSTEM ERROR, FILE NOT COPIED."; 00019000
GO TO EXIT;                           00019100
EXIT:                                 00019200
SV:=DI; SI:=LOC SV; DI:=APTR; DS:=WDS; 00019300
END STREAM PROCEDURE MESSAGE;        00019400
%*****00019500
REAL PROCEDURE TIMECONV;              00019600
%*****00019700
BEGIN                                00019800
INTEGER HRS,MIN,TIM;                 00019900
REAL STREAM PROCEDURE ST(H,M,X);     00020000
BEGIN                                00020100
DI:=LOC ST; DI:=DI+1; SI:=H; DS:=2 DEC; DS:=LIT"!"; 00020200
SI:=M; DS:=2DEC; SI:=X; SI:=SI+6; DS:=2CHR; 00020300
END STREAM ST;                       00020400
TIM := TIME(1);                       00020500
HRS := TIM DIV 216000;                00020600
MIN := (TIM DIV 3600) MOD 60;         00020700
TIM := IF HRS GTR 11 THEN "PM" ELSE "AM"; 00020800
IF HRS GTR 12 THEN HRS := HRS - 12;  00020900

```

35LIT" ———— 11

Data Documents, Inc

```

IF HRS LSS 1 THEN HRS := 12;                                00021000
TIMECONV := ST(HRS,MIN,TIM);                                00021100
END TIMECONV;                                              00021200
*****00021300
1 STREAM PROCEDURE REFORMAT(A,ASEQ);                          00021400
2 BEGIN                                                       00021500
3   DI:=A; DS:=16LIT" ";                                       00021600
4   SI:=ASEQ; DI:=A; DI:=DI+7; DS:=8CHR; DI:=DI-8; DS:=7FILL; 00021700
5   END STREAM PROCEDURE REFORMAT;                             00021800
6 *****00021900
7 PROCEDURE EXPAND(A,RSIZE); VALUE RSIZE; REAL RSIZE; ARRAY A[0]; 00022000
8 *****00022100
9 BEGIN                                                       00022200
10  SAVE ARRAY B[0:RSIZE];                                       00022300
11  DEFINE FTC=SI:=SI+3; DI:=DI+5; SKIP 3DB;                    00022400
12  15(IF SB THEN DS:=SET ELSE DS:=RESET; SKIP SB);            00022500
13  DEFINE CTF=SI:=SI+5; SKIP 3SB; DI:=DI+3;                    00022600
14  15(IF SB THEN DS:=SET ELSE DS:=RESET; SKIP SB);            00022700
15  STREAM PROCEDURE CHANGEVECTORS(A,B);                       00022800
16  BEGIN LOCAL MOMA,MOMB;                                       00022900
17    SI:=LOC A; DI:=LOC MOMA; FTC;                               00023000
18    SI:=LOC B; DI:=LOC MOMB; FTC;                               00023100
19    SI:=LOC MOMB; DI:=LOC A; CTF;                               00023200
20    SI:=LOC MOMA; DI:=LOC B; CTF;                               00023300
21    SI:=LOC B; DI:=MOMA; DS:=WDS;                               00023400
22    SI:=LOC A; DI:=MOMB; DS:=WDS;                               00023500
23    END STREAM PROCEDURE CHANGEVECTORS;                       00023600
24  CHANGEVECTORS(A,B);                                         00023700
25  END PROCEDURE EXPAND;                                       00023800
26 *****00023810
27 PROCEDURE PARITYERROR(RECORD); VALUE RECORD; REAL RECORD;  00023815
28 *****00023820
29 BEGIN                                                       00023825
30  % CALLS ON ALGOL-READ INTRINSIC TO OBTAIN PAR=NO LABEL MESSAGE 00023830
31  % SO THAT "HELP/DISK" PROGRAM WILL BE CALLED                 00023835
32  FILE DISK DISK RANDOM (2,RPB,HDR0,[15:15]);                 00023840
33  FILL DISK WITH MFID,FID;                                     00023845
34  DO BEGIN                                                     00023850
35    READ(DISK[RECORD],RPB,A[*]); RECORD:=RECORD+1;            00023855
36  END UNTIL FALSE;                                           00023860
37  END PROCEDURE PARITYERROR;                                   00023865
38 *****00023900
39 PROCEDURE DISKIO(BUFF,SIZE,DISKADDR,RESULT,MASK,WATE);      00024000
40 *****00024100
41 VALUE BUFF,SIZE,DISKADDR,MASK,WATE;                          00024200
42 INTEGER SIZE,DISKADDR,BUFF; REAL RESULT,MASK; BOOLEAN WATE; 00024300
43 BEGIN                                                       00024400
44  PROCEDURE IOREQUEST(FINAL,IODESC,LOCATION);                00024500
45  VALUE FINAL,IODESC,LOCATION; REAL FINAL,IODESC,LOCATION; 00024600
46  COMMUNICATE(41);                                           00024700
47  REAL UNIT,TINU,IOC,DSKADRS;                                  00024800
48  REAL STREAM PROCEDURE ADRS(BUFF,DISKADDR); VALUE BUFF,DISKADDR; 00024900
49  BEGIN                                                       00025000
50    SI:=LOC DISKADDR; DI:=BUFF; DS:=8DEC;                     00025100
51    SI:=LOC DISKADDR; DI:=LOC ADRS; DS:=8DEC;                 00025200
52  END STREAM PROCEDURE ADRS;                                   00025300
53  IF BOOLEAN( (DSKADRS:=ADRS(BUFF,DISKADDR)),[1:5]) THEN    00025400
54  BEGIN UNIT:=19; TINU:=12; END ELSE BEGIN UNIT:=18; TINU:=6; END; 00025500
55  IOI:=BUFF&SIZE[8:38:10]&                                  00025600
56  ((SIZE,[38:10]+29) DIV 30+512)[18:33:15]&1[24:47:1]&TINU[3:43:5]; 00025700
57

```

```

RESULT:=ADDRESS(RESULT);
IOREQUEST(-IBD&3"347"[25:40:8],IOD,RESULT&UNIT[12:42:6]&1[2:47:1]);
IF WATE THEN WAIT(ADDRESS(RESULT),MASK);
END PROCEDURE DISKIO;
*****
BOOLEAN PROCEDURE GETRECORD(RECORD,CEREAL); VALUE RECORD,CEREAL;
*****
REAL RECORD; BOOLEAN CEREAL;
BEGIN
% READS DISK FILE RECORD INTO ARRAY "A"
OWN REAL NEXTRECORD,RMIN1,RMIN2,RMAX1,RMAX2,ADRS1,ADRS2;
REAL BUFFER;
DEFINE BUFFERSIZE=NEXTRECORD#; % TEMPURARY STORAGE
OWN BOOLEAN SETT,VALID,OK1,OK2;
BOOLEAN FILL1,FILL2;
INTEGER OFFSET;
LABEL START,TRANSFER,EXIT;
%.....
STREAM PROCEDURE MOVE(N,A,B); VALUE N,A;
BEGIN LOCAL T;
SI:=LOC N; SI:=SI+6; DI:=LOC T; DI:=DI+7; US:=CHR;
SI:=A; DI:=B; T(2(DS:=32WDS)); DS:=N WDS;
END STREAM PROCEDURE MOVE;
%.....
PROCEDURE FILLBUFFER(BUFF,RECORD,RESULT,RMIN,RMAX); VALUE BUFF,RECORD;
%.....
REAL BUFF,RECORD,RESULT,RMIN,RMAX;
BEGIN LABEL EXIT;
REAL BLOCK,ROW;
INTEGER SEGMENT,BLOCKADDRESS;
ROW:=(SEGMENT:=(BLOCK:=RECORD DIV RPB)*SPB) DIV SPR;
IF ROW LEQ 19 THEN
IF (BLOCKADDRESS:=DISKROWS[ROW]) NEQ 0 THEN
BEGIN
BLOCKADDRESS:=BLOCKADDRESS+(SEGMENT MOD SPR);
DISKIO(BUFF,(RPB*RSIZE),BLOCKADDRESS,RESULT,MASK,FALSE);
RMAX:=(NEXTRECORD:=(RMIN:=BLOCK*RPB)+RPB)-1;
GO TO EXIT;
END;
RESULT:=MASK; RMIN:=RMAX:=-1;
EXIT:
END PROCEDURE FILLBUFFER;
%.....
IF NOT SETT THEN
BEGIN
IF (BUFFERSIZE:=RPB*RSIZE) GTR 300 THEN
BEGIN
IF BUFFERSIZE GTR 1023 THEN GO TO ERRER;
EXPAND(BUFFER1,BUFFERSIZE);
EXPAND(BUFFER2,BUFFERSIZE);
END;
IF RSIZE GTR 29 THEN
BEGIN
EXPAND(A,RSIZE+1);
A[0]:=0; APTR:=ASTART:=ADDRESS(A[0]);
END;
RMIN1:=RMIN2:=RMAX1:=RMAX2:= -1;
SETT:=TRUE;
END; % OF INITIALIZE
IF RECORD GTR EOFPTR OR RECORD LSS 0 THEN GO TO EXIT;

```

```

00025800
00025900
00026000
00026100
00026200
00026300
00026400
00026500
00026600
00026700
00026800
00026900
00027000
00027100
00027200
00027300
00027400
00027500
00027600
00027700
00027710
00027720
00027730
00027800
00027900
00028000
00028100
00028200
00028300
00028400
00028500
00028600
00028700
00028800
00028900
00029000
00029100
00029200
00029300
00029400
00029500
00029600
00029700
00029800
00029900
00030000
00030100
00030110
00030200
00030300
00030400
00030500
00030600
00030700
00030800
00030900
00031000
00031100
00031200
00031300

```

```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57

```

1	IF NOT OK1 THEN	00031400
2	BEGIN	00031500
3	BUFFER1[0]:=0; ADRS1:=ADDRESS(BUFFER1[0]); OK1:=TRUE;	00031600
4	END;	00031700
5	IF CEREAL AND NOT OK2 THEN	00031800
6	BEGIN	00031900
7	BUFFER2[0]:=0; ADRS2:=ADDRESS(BUFFER2[0]); OK2:=TRUE;	00032000
8	END;	00032100
9	IF CEREAL AND NOT VALID THEN FILL2:=TRUE; % READING SERIALY	00032200
10	VALID:=CEREAL;	00032300
11	START:	00032400
12	IF RECORD GEQ RMIN1 AND RECORD LEQ RMAX1 THEN	00032500
13	BEGIN	00032600
14	BUFFER:=ADRS1;	00032700
15	IF NOT BOOLEAN(RESULT1.[19:1]) THEN WAIT(ADDRESS(RESULT1),MASK);	00032800
16	IF RESULT1.[28:2] NEQ 0 THEN PARITYERROR(RMIN1);	00032810
17	FILL1:=(RECORD=RMAX1) AND CEREAL;	00032900
18	GO TO TRANSFER;	00033000
19	END;	00033100
20	IF CEREAL THEN IF RECORD GEQ RMIN2 AND RECORD LEQ RMAX2 THEN	00033200
21	BEGIN	00033300
22	BUFFER:=ADRS2;	00033400
23	IF NOT BOOLEAN(RESULT2.[19:1]) THEN WAIT(ADDRESS(RESULT2),MASK);	00033500
24	IF RESULT2.[28:2] NEQ 0 THEN PARITYERROR(RMIN2);	00033510
25	FILL2:=(RECORD=RMAX2) AND CEREAL;	00033600
26	GO TO TRANSFER;	00033700
27	END;	00033800
28	IF NOT BOOLEAN(RESULT1.[19:1]) THEN WAIT(ADDRESS(RESULT1),MASK);	00033900
29	IF RESULT1.[28:2] NEQ 0 THEN PARITYERROR(RMIN1);	00033910
30	FILLBUFFER(ADRS1,RECORD,RESULT1,RMIN1,RMAX1);	00034000
31	IF CEREAL THEN	00034100
32	BEGIN	00034200
33	IF NOT BOOLEAN(RESULT2.[19:1]) THEN WAIT(ADDRESS(RESULT2),MASK);	00034300
34	IF RESULT2.[28:2] NEQ 0 THEN PARITYERROR(RMIN2);	00034310
35	FILLBUFFER(ADRS2,NEXTRECORD,RESULT2,RMIN2,RMAX2);	00034400
36	END;	00034500
37	FILL2:=FALSE;	00034600
38	GO TO START;	00034700
39	TRANSFER:	00034800
40	OFFSET:=(RECORD MOD RPB)*RSIZE+1;	00034900
41	MOVE(RSIZE,BUFFER+OFFSET,A[2*REAL(FORMT)]);	00035000
42	GETRECORD:=TRUE;	00035100
43	IF FILL1 THEN	00035200
44	BEGIN	00035300
45	IF NOT BOOLEAN(RESULT1.[19:1]) THEN WAIT(ADDRESS(RESULT1),MASK);	00035400
46	IF RESULT1.[28:2] NEQ 0 THEN PARITYERROR(RMIN1);	00035410
47	FILLBUFFER(ADRS1,NEXTRECORD,RESULT1,RMIN1,RMAX1);	00035500
48	END;	00035600
49	IF FILL2 THEN	00035700
50	BEGIN	00035800
51	IF NOT BOOLEAN(RESULT2.[19:1]) THEN WAIT(ADDRESS(RESULT2),MASK);	00035900
52	IF RESULT2.[28:2] NEQ 0 THEN PARITYERROR(RMIN2);	00035910
53	FILLBUFFER(ADRS2,NEXTRECORD,RESULT2,RMIN2,RMAX2);	00036000
54	END;	00036100
55	EXIT:	00036200
56	END PROCEDURE GETRECORD;	00036300
57	%*****	00036400
58	PROCEDURE BSEARCH(LREC,LRANGE); REAL LREC,LRANGE;	00036500
59	%*****	00036600
60	% LREC IS RELATIVE DISK ADDRESS OF FILE RECORD	00036700

```

% L RANGE IS SEQUENCE NUMBER OF RECORD WE ARE SEARCHING FOR
BEGIN
REAL ADJ, REC, SEQ;
LABEL START, EXIT;
- REC:=LREC+(EOFPTR-LREC) DIV 2; % CENTER OF RANGE
GO TO START;
DO BEGIN
IF L RANGE LSS SEQ THEN % MOVE BACK
REC:=(REC:=REC-ADJ)*REAL(REC.[1:1]=0) ELSE
IF REC:=(LREC:=REC)+ADJ GTR EOFPTR THEN REC:=EOFPTR;
START:
IF GETRECORD(REC, FALSE) THEN
IF (SEQ:=INPCONV(A[SEQLOC]))=L RANGE THEN
BEGIN LREC:=REC; GO TO EXIT; END;
ADJ:=(REC-LREC) DIV 2;
END UNTIL ADJ LEQ 1;
EXIT:
END PROCEDURE BSEARCH;
%=====
A[0]:=0; APTR:=ASTART:=ADDRESS(A[0]);
DISKWAIT(1, A, 30, COMMON); COMMON:=0; % ESP DISK RECORD
DATAFILE := BOOLEAN(A[1].[3:1]); % TYPE DATA FILE
WORKFILE := BOOLEAN(A[1].[6:1]); % INPUT FILE=WORKFILE
CONCISE := BOOLEAN(A[1].[8:1]); % CONCISE OPTION SET
USER := A[2]; % USER CODE
MFID := A[3]; % FIRST NAME OF INPUT FILE
FID := A[4]; % SECOND NAME OF INPUT FILE
OUTPUTNAME := A[9]; % NAME OF OUTPUT FILE
OUTCONV(LINE, A[1].[40:8], FALSE); % DECIMAL VALUE OF LINE NUMBER
IF SEQCHECK:=(NPARAMS:=A[1].[27:6]) GTR 0 THEN % PARAMETERS GIVEN
MOVE(9, A[21], PARAMS[*]);
IF RESEQ := A[5] NEQ 0 THEN % RESEQUENCING PARAMTERS GIVEN
BEGIN
STARTRESEQ := A[5];
ENDRESEQ := A[6];
INCREMENT := A[8];
BASE := A[7]-INCREMENT;
END;
PRINTER := OUTPUTNAME.[6:6]="2"; % COPY TO PRINTER
PUNCH := OUTPUTNAME.[6:6]="3"; % COPY TO PUNCH
A[0]:= -1; GETHEADER(A, MFID, FID, USER);
IF A[0] LSS 0 THEN
BEGIN
MESSAGE(APTR, 5); % SYSTEM ERROR
TWXOUT(A[0], MSGSIZE, 1);
COMMON := 0 & 1[2:47:1]; % ERROR FLAG FOR CANDE
GO TO EXIT;
END;
HDRO := A[0];
PSIZE:=RSIZE:= HDRO.[1:14];
RPB := HDRO.[30:12];
SPB := HDRO.[42:6];
FILETYPE := A[4].[36:6];
EOFPTR := A[7];
SPR := A[8];
MOVE(20, A[10], DISKROWS[0]);

```

```

00036800
00036900
00037000
00037100
00037200
00037300
00037400
00037500
00037600
00037700
00037800
00037900
00038000
00038100
00038200
00038300
00038400
00038500
00038600
00038700
00038800
00038900
00039000
00039100
00039200
00039300
00039400
00039500
00039600
00039700
00039800
00039900
00040000
00040100
00040200
00040300
00040400
00040500
00040600
00040700
00040800
00040900
00041000
00041100
00041200
00041300
00041400
00041500
00041600
00041700
00041800
00041900
00042000
00042100
00042200
00042300
00042400
00042500
00042600
00042700

```

real WHICH PRNT

(LPA) (LPB)
 ← 2 = "5" or "6"

WHICH PRNT =
 (if Xφ = "2" then 0 else
 if Xφ = "5" then 1 else 2) #
 Xφ = outputname.[6:6] #

Data Documents/Inc.

```

IF NOT SEQCHECK AND ECFPTR GTR 29 THEN HRANGE:=EUFMARK; * SERIAL READ 00042800
IF RSIZE NEQ 10 AND ECFPTR GEQ 0 THEN * NON-STANDARD RECORD SIZE 00042900
IF (PUNCH AND RSIZE GTR 10 ) OR OUTPUTNAME.[6:12]="1S" THEN 00043000
BEGIN 00043100
ERROR: MESSAGE(APTR,0); * NON-STANDARD FILE ERROR 00043200
TWXOUT(A[0],MSGSIZE,1); 00043300
COMMON:=0 & 1[2:47:1]; * ERROR FLAG FOR CANDE 00043400
GO TO EXIT; 00043500
END; 00043600

IF WORKFILE AND NOT DATAFILE THEN SFLG:=TRUE * INPUT FILE=WORKFILE 00043700
ELSE SFLG:=(RSIZE=10 AND FILETYPE NEQ 8); * 10 WORD,SEQUENTIAL FILE 00043800

IF PRINTER AND SFLG AND FILETYPE=10 THEN * INFO FILE 00043900
BEGIN 00044000
PSIZE:=RSIZE-1; SEQLOC:=9; 00044100
END 00044200
ELSE SEQLOC:=IF (FORMT:=SFLG AND PRINTER) THEN 11 ELSE 9; 00044300

RESULT1 := RESULT2 := MASK; * INITIALIZE RESULT DESCRIPTOR 00044400

IF MAKETAB:=(OUTPUTNAME.[6:12]="1S" AND NOT DATAFILE) THEN 00044500
BEGIN 00044600
FILL NEWTAB WITH " "&"1T"[6:36:12]&LINE[18:30:18],USER; 00044700
I:=600; * RECORDS PER ROW 00044800
J:=300; * WORDS PER BLOCK 00044900
END 00045000
ELSE 00045100
BEGIN 00045200
I:=(((HORO.[30:12] / HORO.[42:6]) * SPR) + 0.5) DIV 1; * RECORDS/ROW 00045300
J:=HORO.[15:15]; * WORDS PER BLOCK 00045400
END; 00045500

BEGIN * INNER BLOCK 00045600
FILE OUTFILE DISK SERIAL [20:1] (2,RSIZE,J,SAVE 30); 00045700
LABEL NEXTPARAM, READIN, FINISHED; 00045800

IF (PRINTER OR PUNCH) THEN * COPY TO PRINTER OR COPY TO PUNCH 00045900
BEGIN 00046000
MESSAGE(APTR,(IF PRINTER THEN 3 ELSE 4)); 00046100
PLACESTRING(APTR,"/"&OUTPUTNAME[6:12:36],FALSE); * FIRST NAME 00046200
PLACESTRING(APTR,FID,FALSE); * SECOND NAME 00046300
PLACESTRING(APTR,TIME(6)&" "[6:42:6],FALSE); * DAY OF WEEK 00046400
I:="0/00/00"&(I:=TIME(5))[1:13:11]&I[18:24:12]&I[36:36:12]; * DATE 00046500
IF I.[1:5] NEQ 0 THEN 00046600
BEGIN 00046700
PLACESTRING(APTR,I.[1:5],TRUE); 00046800
IF (J:=APTR.[30:3]) GTR 0 THEN APTR.[30:3]:=(J-1) ELSE 00046900
APTR:=(APTR.[33:15]-1)&7[30:45:3]; 00047000
END; 00047100
PLACESTRING(APTR,I,FALSE); 00047200
PLACESTRING(APTR,I:=TIMECONV,FALSE); * TIME OF DAY 00047300
PLACESTRING(APTR," ",FALSE); 00047310
I:=(MSGSIZE) DIV 8; APTR:=ASTART; 00047320
IF PRINTER THEN WRITE(PRNT[DBL],I,A[*]); ELSE WRITE(PNCH,I,A[*]); 00047330
IF PRINTER THEN WRITE(PRNT[DBL]); 00047340
END * IF PRINTER OR PUNCH 00047350
ELSE FILL OUTFILE WITH OUTPUTNAME,USER; 00047360
IF ECFPTR LSS 0 THEN GO TO FINISHED; 00047400
00047500
00047600
00047700
00047800
00047900
00048000
00048100

```

PRNT[DBL]
to PRNTFILES[WHICH PRNT][DBL]

Data Document 171c


```

PREVSEQ := NCT := -1; LREC := 0; 00048200
IF SEQCHECK THEN % SEQUENCE RANGE SPECIFIED 00048300
BEGIN 00048400
  NPARAMS:=NPARAMS-1; 00048500
  SRCESEQ:=PC:=-1; 00048600
  LREC:=1; 00048700
NEXTPARAM: 00048800
  LREC:=LREC-1; 00048900
  LRANGE:=HRANGE:=PARAMS[PC:=PC+1].[21:27]; 00049000
  IF LRANGE=EOFMARK THEN % "END" 00049100
  BEGIN 00049200
    LREC:=EOFPTR; LRANGE:=0; GO TO READIN; 00049300
  END; 00049400
  IF PC LSS NPARAMS THEN % MORE PARAMETERS SPECIFIED 00049500
  IF BOOLEAN(PARAMS[PC+1].[1:1]) THEN 00049600
  HRANGE:=PARAMS[PC:=PC+1].[21:27]; 00049700
  IF NOT SFLG THEN % PARAMETER IS RECORD LOCATION 00049800
  BEGIN 00049900
    LREC := (LRANGE-1)*REAL(LRANGE NEQ 0); 00050000
    GO TO READIN; 00050100
  END; 00050200
  IF SRCESEQ=LRANGE THEN % ALREADY HAVE THIS RECORD 00050300
  BEGIN 00050400
    GO TO READIN; 00050500
  END; 00050600
  IF (EOFPTR=LREC) GTR 30 THEN BSEARCH(LREC,LRANGE); % BINARY SEARCH 00050700
  END; % IF PARAMETERS SPECIFIED 00050800
READIN: 00050900
  IF GETRECORD(LREC,(HRANGE GTR LRANGE)) THEN 00051000
  BEGIN 00051100
    LREC:=LREC+1; 00051200
    SRCESEQ:=IF SFLG THEN INPCONV(A[SEQLOC]) ELSE LREC; 00051300
    IF SEQCHECK THEN % CHECK SEQUENCE RANGE 00051400
    BEGIN 00051500
      IF SRCESEQ LSS LRANGE THEN GO TO READIN; % TOO LOW 00051600
      IF SRCESEQ GTR HRANGE THEN 00051700
      IF PC LSS NPARAMS THEN GO TO NEXTPARAM ELSE GO TO FINISHED; 00051800
    END; % SEQUENCE CHECKING 00051900
    NCT:=NCT+1; % OUTPUT FILE RECORD COUNT 00052000
    OUTSEQ:= 00052100
    IF MAKETAB AND NOT SFLG THEN INPCONV(A[SEQLOC]) ELSE SRCESEQ; 00052200
    IF RESEQ THEN % RESEQUENCING THE FILE 00052300
    IF OK OR SRCESEQ GEQ STARTRESEQ THEN 00052400
    IF RESEQ:=OK:=SRCESEQ LEQ ENDRESEQ THEN % IN RANGE 00052500
    BEGIN 00052600
      OUTSEQ:=BASE:=BASE+INCREMENT; % NEW SEQUENCE NUMBER 00052700
      OUTCONV(A[SEQLOC],OUTSEQ,FALSE); % NEW NUMBER IN RECORD 00052800
    END; % IF RESEQUENCING 00052900
    IF MAKETAB THEN % MAKE SURE SEQUENCE NUMBERS ARE IN ORDER 00053000
    BEGIN 00053100
      IF OUTSEQ LEQ PREVSEQ THEN % OUT OF SEQUENCE 00053200
      BEGIN 00053300
        ERROR:=NCT BRAAK; % DONT PRINT ERROR IF BREAK RECEIVED 00053400
        ERRSEQ:=OUTSEQ; 00053500
        OUTSEQ:=PREVSEQ+2; % ADJUST SEQUENCE NUMBER 00053510
        OUTCONV(A[SEQLOC],OUTSEQ,FALSE); 00053600
      END; 00053700
    IF NPTR:=NPTR+1 GTR 29 THEN % TAB SEGMENT IS FILLED 00053800
    BEGIN 00053900
      WRITE(NEWTAB,30,N[*]); 00054000

```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57

Data Documents Inc.

```

      NPTR:=0;
      END;
      N[NPTR]:=0 & NCT[4:32:16] & OUTSEQ[21:21:27];
      END; % IF MAKETAB
      IF FORMT THEN REFORMAT(A,A[SEWLOC]);
      IF PRINTER THEN
      BEGIN
      J:=PSIZE+REAL(FORMT);
      IF NOT SFLG AND RSIZE=10 THEN % DISPLAY RECORD NUMBER
      BEGIN
      OUTCONV(A[J],LREC,TRUE);
      J:=J+1;
      END;
      WRITE(PRNTIDBL,J,A[*]);
      END % IF PRINTER
      ELSE IF PUNCH THEN WRITE(PNCH,RSIZE,A[*]);
      ELSE WRITE(OUTFILE,RSIZE,A[*]);
      IF ERROR THEN
      BEGIN
      MESSAGE(APTR,2); % SEQUENCE ERROR
      PLACESTRING(APTR,PREVSEQ,TRUE);
      PLACESTRING(APTR,ERRSEQ,TRUE);
      TWXOUT(A[0],MSGSIZE,1);
      ERROR:=FALSE;
      END;
      PREVSEQ:=OUTSEQ;
      GO TO READIN;
      END; % IF MORE RECORDS IN FILE
      FINISHED:
      IF MAKETAB THEN % CHECK LAST ENTRY IN TAB FILE
      BEGIN
      IF NPTR:=NPTR+1 GTR 29 THEN % FULL TAB SEGMENT
      BEGIN
      WRITE(NEWTAB,30,N[*]);
      NPTR:=0;
      END;
      N[NPTR]:=EOFMARK;
      WRITE(NEWTAB,30,N[*]);
      END; % IF MAKETAB
      IF NOT (PRINTER OR PUNCH) THEN
      BEGIN
      IF NCT GEQ 0 THEN PLACETYPE(FILETYPE,OUTFILE);
      LOCK(OUTFILE,*);
      IF MAKETAB THEN LOCK(NEWTAB,*);
      END;
      END INNER BLOCK;
      IF NOT CONCISE THEN
      BEGIN
      PLACESTRING(APTR,NCT+1,TRUE);
      MESSAGE(APTR,1); % NUMBER OF RECORDS COPIED
      PLACESTRING(APTR,OUTSEQ,TRUE);
      TWXOUT(A[0],MSGSIZE,1);
      END; % IF NOT CONCISE
      COMMON:=COMMON & (NCT+1)[3:3:45]; % PASS RECORD COUNT TO CANDE
      IF NOT BOOLEAN(RESULT1.[19:1]) THEN WAIT(ADDRESS(RESULT1),MASK);
      IF NOT BOOLEAN(RESULT2.[19:1]) THEN WAIT(ADDRESS(RESULT2),MASK);
      EXIT;
      END PROGRAM.
      END;END.          LAST CARD ON OCRDING TAPE
      MIT + GETAREA(0),[FF] + RES;

```

```

00054100
00054200
00054300
00054400
00054500
00054600
00054700
00054800
00054900
00055000
00055100
00055200
00055300
00055400
00055500
00055600
00055700
00055800
00055900
00056000
00056100
00056200
00056300
00056400
00056500
00056600
00056700
00056800
00056900
00057000
00057100
00057200
00057300
00057400
00057500
00057600
00057700
00057800
00057900
00058000
00058100
00058200
00058300
00058400
00058500
00058600
00058700
00058800
00058900
00059000
00059100
00059200
00059300
00059400
00059500
00059600
00059700
00059800
99999999
03056500

```

← PRINT FILES [WHICH PRINT]

LABEL 0000000COPRINTER0017510000 EX OBJECT/READ;FILE SOURCEFILE=SYMBOL/COPY;END+

OBJECT /READ

1		1
2		2
3		3
4		4
5		5
6		6
7		7
8		8
9		9
10		10
11		11
12		12
13		13
14		14
15		15
16		16
17		17
18		18
19		19
20		20
21		21
22		22
23		23
24		24
25		25
26		26
27		27
28		28
29		29
30		30
31		31
32		32
33		33
34		34
35		35
36		36
37		37
38		38
39		39
40		40
41		41
42		42
43		43
44		44
45		45
46		46
47		47
48		48
49		49
50		50
51		51
52		52
53		53
54		54
55		55
56		56
57		57

Data Documents/Inc