

SYMBOL / MAKCAST

Date Documents, 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	

BEGIN COMMENT MCP SYMBOLIC SUBPROGRAM TAPE MAINTENANCE
UTILITY ROUTINE;

COMMENT: * TITLE: B5500/B5700 MARK XIV SYSTEM RELEASE *
 * FILE ID: SYMBOL/MAKCAST TAPE ID: SYMBOL1/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 AA393180 *;

BOOLEAN ARRAY OPEN[0:2];
 BOOLEAN SING;
 BOOLEAN PATCHTOG, PATCHCTOG;
 DEFINE PFRQMB=IF MLISTOG OR LISTOG THEN PPP.[47:1]+1;
 IF MPUNTOG OR PUNTOG THEN PPP.[46:1]+1#;
 DEFINE PFROMM=IF MLISTOG THEN PPP.[47:1]+1;
 IF MPUNTOG THEN PPP.[46:1]+1#;
 INTEGER NEWRC, EDROW, EDPOS, IMGOUT, BCOU;
 INTEGER IMG, REL, RC;
 BOOLEAN CCBUFF;
 BCCLEAN B1 ;
 INTEGER ARRAY RECC[0:2];
 ARRAY FINARR[0:55];
 INTEGER ROW;
 INTEGER RR,11;
 BOOLEAN ENDTOG;
 BOOLEAN RENAMETOG;
 BOOLEAN LISTOG,PUNTOG, SEQATOG, SEQCTOG, WITHTOG, AFTERTOG;
 INTEGER SEQINC,WITHINDEX;
 SAVE ARRAY TESTARR [0:26];
 REAL DUMY; % DO NOT REMOVE THIS CARD
 SAVE FILE CASTA DISK SERIAL [20:100] "CASTA" "LIBRARY"(1,56,SAVE 2);
 SAVE FILE CASTB DISK SERIAL [20:100] "CASTB" "LIBRARY"(1,56,SAVE 2);
 SAVE FILE CASTC DISK SERIAL [20:100] "CASTC" "LIBRARY"(1,56,SAVE 2);
 FILE IN Z "OCDIMG" (1,56,10), CARD(1,10), SOLT(1,56,10);
 FILE IN A DISK SERIAL "CASTA" "LIBRARY"(1,56);
 FILE IN B DISK SERIAL "CASTB" "LIBRARY"(1,56);
 FILE IN C DISK SERIAL "CASTC" "LIBRARY"(1,56);
 FILE OUT LINE 4(2,15); ARRAY PRINT [0:14];
 DEFINE PRINTLINE = WRITE(LINE,15,PRINT[*])#;
 FILE OUT PUNCH 0 (1,10);
 SWITCH FILE FOUT+CASTA,CASTB,CASTC ;
 SWITCH FILE FIN+ A,B,C,Z,CARD,SOLT ;
 SAVE FILE CFA "CASTA"(1,56,SAVE 100);
 CFB "CASTB"(1,56,SAVE 100);
 CFC "CASTC"(1,56,SAVE 100);
 SWITCH FILE CF + CFA,CFB,CFC;
 BOOLEAN COPYTOG;
 SAVE ARRAY DUMMY[0:9], PATCHDUMMY[0:9];
 INTEGER GT5;
 DEFINE TEST1= IF GT5#0 THEN IF GT5=1 THEN GO TO E10 ELSE GO TO E9 #;
 INTEGER ADDINDEX;
 INTEGER X;
 INTEGER TEMP, LIBOUTIX, LIBINIX;
 LABEL MORE, LASTCARD;
 LABEL E0, E1, E2, E3, E4, E5, E6, E7, E8, E9, E10;

00000100
 00000200
 00000300
 00000310
 00000311
 00000312
 00000313
 00000314
 00000315
 00000316
 00000317
 00000318
 00000319
 00000400
 00000500
 00000600
 00000700
 00000800
 00000900
 00001000
 00001100
 00001200
 00001300
 00001400
 00001500
 00001600
 00001650
 00001700
 00001800
 00001900
 00002000
 00002100
 00002200
 00002300
 00002310
 00002400
 00002500
 00002505
 00002510
 00002515
 00002600
 00002700
 00002800
 00002850
 00002900
 00003000
 00003100
 00003200
 00003300
 00003400
 00003500
 00003600
 00003700
 00003800
 00003900
 00004000
 00004100
 00004200
 00004300
 00004400

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.

```

SAVE ARRAY ACCUMA[0:7], ACCUMB[0:7], ACCUMC[0:7], ACCUMD[0:7], ACCUME[0:9]; 00004500
COMMENT THESE ARRAYS ARE PASSED BY SUBCCHANDLER 00004600
      TO SCAN AND RECEIVE CONTENTS OF SUBCONTROL 00004700
      CARDS. EVENTUALLY, THEY WILL STORE VARIOUS IDENTIFIERS 00004800
      APPEARING ON THE SUBCONTROL CARD. 00004900
1. ACCUMA (AND NA WHICH TELLS HOW MANY CHARS) 00005000
   WILL ALWAYS CONTAIN THE IDENTIFIER ASSOCIATED 00005100
   WITH ADD IDENTIFIER 00005200
   PATCH IDENTIFIER 00005300
   REPLACE IDENTIFIER 00005400
   DELETE IDENTIFIER 00005500
   IDENTIFIER ETC (MAKE OR DISPLAY SUBCC) 00005600
2. ACCUMB (AND NB WHICH TELLS HOW MANY CHARS) 00005700
   WILL ALWAYS CONTAIN THE IDENTIFIER ASSOCIATED 00005800
   WITH AFTER IDENTIFIER 00005900
   WITH IDENTIFIER 00006000
3. ACCUMC (AND NC WHICH TELLS HOW MANY CHARS) 00006100
   WILL ALWAYS CONTAIN THE IDENTIFIER ASSOCIATED 00006200
   WITH RENAME IDENTIFIER; 00006300
INTEGER NA, NB, NC, ND; COMMENT NO. OF CHARS. IN ACCUMA---ACCUMD; 00006400
INTEGER NCR; COMMENT USED BY SUBCCHANDLER-CONTAINS 00006500
      THE ABSOLUTE ADDRESS OF NEXT 00006600
      BUFFER POSITION TO BE SCANNED; 00006700
INTEGER IDENTIX; COMMENT FOR MAKE SUBCC OF TYPE XXX ON B. THIS 00006800
      INDEX LOCATES BLOB WHERE 0=A 00006900
      1=B 00007000
      2=C 00007100
      3=CCRD 00007200
      4=CARD READER; 00007300
INTEGER ZOTTEMP; 00007400
BOOLEAN RETURN; COMMENT USED BY PROCEDURE SUBCCHANDLER 00007500
      TO SIGNAL CONTINUATION CARDS; 00007600
BOOLEAN ADDTOG; COMMENT SET FOR MAKE SUBCC OF TYPE 00007700
      ADD XX ETC.; 00007800
BOOLEAN ADDENDTOG; COMMENT SET FOR MAKE SUBCC OF TYPE 00007900
      ADD XYZ END--OR ADD XYZ ON B END LIST ETC.; 00008000
BOOLEAN ADDAFTERTOG; COMMENT SET FOR MAKE SUBCC OF TYPE 00008100
      ADD XYZ ON B AFTER BLOBX ETC.; 00008200
BOOLEAN PATCHTOG, REPLACETO, DELETEREG; 00008300
      COMMENT THESE THREE AND 00008400
      ADDTOG INDICATE BASIC TYPE OF 00008500
      MAKE FUNCTION TO BE PERFORMED. 00008600
      IF NONE ARE SET, WE CAN ASSUME 00008700
      A SUBCC OF TYPE "XYZ ON B" WAS 00008800
      ENTERED. IDENTIX TELLS WHERE 00008900
      THAT BLOB CAN BE FOUND; 00009000
BOOLEAN MLISTOG, MPUNTOG, DIRTOG, DISTOG, MKTOG, MKFTOG, 00009100
      MERRIOG; 00009200
SWITCH FORMAT SWF+ (" "), 00009300
  ("***** JOB TERMINATED. ERROR IN MASTER CONTROL CARD."), 00009400
  ("***** JOB TERMINATED. ERROR IN SUBCONTROL CARD."), 00009500
  ("***** JOB COMPLETE - NO ERROR SITUATIONS ENCOUNTERED"), 00009600
  ("***** JOB TERMINATED- ITEM NOT FOUND IN DIRECTORY"), 00009700
  ("***** JOB TERMINATED-DIRECTORY ON TAPE HAS IMPROPER FORMAT" 00009800
  ), ("***** JOB TERMINATED-THREE ATTEMPTS TO POSITION THE TAPE WE 00009900
  RE UNSUCCESSFUL"), 00010000
  ("***** JOB TERMINATED. MAKE FUNCTION DELETE OPTION 00010100
  NO INPUT TAPE SPECIFIED"), 00010200
  ("***** JOB TERMINATED. PROGRAM NOT PRESENT IN DIREC 00010300
  TORY"), 00010400

```

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 Document, Inc.

```

("***** JOB TERMINATED. REFERENCED PROGRAM NOT FORWARD ON MASTER INPL00010500
T TAPE."),
("***** JOB TERMINATED. OUTPUT TAPE DIRECTORY OVERFLOW"),
("***** JOB TERMINATED. ADD TYPE DIRECTORY OVERFLOW");
STREAM PROCEDURE FIXFIB(A,B);
  VALUE B;
  BEGIN
  LOCAL X,Y;
  LABEL EXIT;
  SI ← A; SI ← SI+8; B(SI+SI+8);
  DI ← LOC X; DS ← WDS; SI ← X; SI ← SI+16;
  DI ← LOC X; DS ← WDS; SI ← X; SI ← SI+40;
  X ← SI; SI ← SI+40; SI ← SI+48; SKIP 7 SB;
  IF SB THEN BEGIN DI ← X; SKIP 43 DB;
  DS ← SET; GO EXIT END;
  EXIT: END FIXFIB;
PROCEDURE PAG;
  BEGIN
  IF NOT SING THEN WRITE(LINE(PAGE));
  END PAG;
BOOLEAN STREAM PROCEDURE LOOKUP(A,T,TEMP,N,M); VALUE M;
BEGIN LOCAL X;
  SI ← A; DI ← T; M(DI+DI+8); IF 7 SC ≤ DC THEN
  BEGIN TALLY ← 1; LOOKUP ← TALLY; SI ← SI-7; DI ← DI-7; IF 7 SC = DC
  THEN BEGIN X ← DI; SI ← X;
  DI ← N; DI ← DI+7; DS ← CHR; DI ← TEMP; DI ← DI+7; DS ← 1 LIT "1";
  END END;
  COMMENT A IS ACCUM, T IS TESTARR, TEMP IS BOOLEAN FOR EQUALITY
  FOUND, N IS NUOUT-VALUE, M IS MIDDLE;
END LOOKUP;
BOOLEAN STREAM PROCEDURE LEGALNUM(A,N);
BEGIN LABEL L,F,EXIT; LOCAL T; SI ← N; DI ← LOC T; DS ← WDS;
TALLY ← 1; LEGALNUM ← TALLY; SI ← A; IF SC = "1" THEN BEGIN
SI ← SI+1; (IF SC = "0" THEN BEGIN SI ← SI+1; GO TO L; END
ELSE JUMP OUT 1 TO F; L:); END ELSE GO TO F;
TALLY ← T; TALLY ← TALLY+1; T ← TALLY;
COMMENT NOW INPUT CONVERT T CHRS;
DI ← N; SI ← A; DS ← T OCT; GO TO EXIT;
F: TALLY ← 0; LEGALNUM ← TALLY;
EXIT: END LEGALNUM;
BOOLEAN STREAM PROCEDURE LEGALID(A,N); VALUE N;
BEGIN LABEL EXIT,BAD,B,F;
SI ← A; IF SC = ALPHA THEN GO B ELSE GO BAD;
B: N(IF SC = ALPHA THEN BEGIN
IF SC = "0" THEN GO F ELSE BEGIN
TALLY ← 1; GO F; END;
END;
IF SC ≠ "-" THEN JUMP OUT TO BAD;
F: SI ← SI+1;);
SI ← SI-1; IF SC = "-" THEN GO BAD ELSE GO
EXIT;
BAD: TALLY ← 0;
EXIT: LEGALID ← TALLY;
END OF LEGALID;
INTEGER PROCEDURE SCANSEARCH(ACCUM,N);
ARRAY ACCUM[0]; INTEGER N; BEGIN
INTEGER LOW,HIGH,MIDDLE,NUOUT;
LABEL EXIT,A;
BOOLEAN TEMP;

```

```

00010600
00010700
00010800
00010820
00010830
00010840
00010850
00010860
00010870
00010880
00010890
00010891
00010892
00010893
00010894
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
00014010
00014020
00014030
00014035
00014040
00014050
00014060
00014100
00014200
00014300
00014400
00014500

```

```

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

```

```

COMMENT SCANSEARCH WILL FIND 0=ERROR                                00014600
                                1 THRU 18 = RESERVED WDS           00014700
                                19= LEGAL IDENT                    00014800
                                20= LEGAL NUMERIC;                00014900
                                IF N>7 THEN GO TO A; HIGH+27; LOW+0; 00015000
WHILE(HIGH-LOW)>1 DO BEGIN MIDDLE+(HIGH+LOW)DIV 2;                 00015100
IF LOOKUP(ACCUM+J,TESTARRL*),TEMP,NUOUT,MIDDLE) THEN BEGIN      00015200
IF TEMP THEN GO TO EXIT ELSE HIGH+MIDDLE END                     00015300
ELSE LOW+MIDDLE; END;                                           00015400
COMMENT VALUE IN ACCUM IS NOT RESERVED WD;                       00015500
A: IF LEGALID(ACCUM,N) THEN BEGIN NUOUT+19; GO TO EXIT; END      00015600
ELSE BEGIN N+N-1;                                               00015700
IF LEGALNUM(ACCUM,N) THEN NUOUT+20 ELSE NUOUT +0; END;          00015800
EXIT: SCANSEARCH+ NUOUT; END SCANSEARCH;                          00015900
STREAM PROCEDURE PRINTCC(LINE, CARD);                              00016000
BEGIN                                                            00016100
DI+LINE; DS+8 LIT " ";                                           00016200
SI+ CARD; DS+ 9 WDS;                                             00016300
DS+ 16 LIT " "; DS + WDS;                                        00016400
2(SI+LINE; DS+ WDS);                                           00016500
END PRINTCC;                                                    00016600
STREAM PROCEDURE ZOT(ZOTTEMP,CHAR,BUFFER);                        00016700
VALUE CHAR;                                                      00016800
BEGIN DI+ZOTTEMP; SI+BUFFER; DS+WDS;                             00016900
DI+BUFFER; SI+LOC BUFFER; SI+SI-1; DS+CHR;                      00017000
END PROCEDURE ZOT;                                              00017100
INTEGER STREAM PROCEDURE MKABS(A);                                00017200
BEGIN DI+A; MKABS+DI; END PROCEDURE MKABS;                       00017300
BOOLEAN STREAM PROCEDURE LEGALCC(NCRN,NCRV,RETURN);              00017400
VALUE NCRV; BEGIN LOCAL T1; LABEL A,B,OUX,C,D;                  00017500
LOCAL TEMP;                                                     00017600
SI+ NCRV; DI+LOC T1; DI+ DI + 2; DS+ 6 LIT "SSSS-S";           00017700
DI+ DI - 6;                                                      00017800
IF 3 SC=DC THEN GO TO A;                                         00017900
SI+SI-3; IF 3 SC=DC THEN GO TO B;                                 00018000
C: TALLY+0; LEGALCC+TALLY; GO TO OUX;                             00018100
A: DI+RETURN; DI+DI+7; DS+ 1 LIT"0";                              00018200
D: TALLY+1; LEGALCC+TALLY; GO TO OUX;                             00018300
B: DI+RETURN; DI+DI+7;DS+1 LIT "1"; GO TO D;                     00018400
OUX: T1+SI; DI+NCRN;                                             00018500
SI+ LOC T1; DS+WDS;                                             00018600
END PROCEDURE LEGALCC;                                           00018700
PROCEDURE READACARD;                                             00018800
BEGIN IF NOT CCHUFF THEN RELEASE(CARD) ELSE                      00018900
CCBUFF+FALSE;                                                  00019000
PRINTCC (PRINT[0], CARD(0)); PRINTLINE;                          00019100
ZOT(ZOTTEMP, "%",CARD(9));                                       00019200
NCR+ MKABS (CARD(0));                                           00019300
END PROCEDURE READACARD;                                         00019400
BOOLEAN STREAM PROCEDURE SCAN(ACCUM,N,NCRN,NCRV);               00019500
VALUE NCRV;                                                      00019600
BEGIN LABEL A, C, RESETNCR, SETNUMCHR, D, OUX;                  00019700
LOCAL TEMP;                                                      00019800
SI+ NCRV; DI+ACCUM;                                             00019900
2(35(IF SC=" " THEN SI+SI+1 ELSE JUMP OUT 2 TO A));              00020000
A: TALLY+1; IF SC="%" THEN BEGIN DS+CHR; GO SETNUMCHR;          00020100
END;                                                             00020200
63(DS+CHR; IF SC="%" THEN JUMP OUT 1 TO C; IF                    00020300
SC=" " THEN JUMP OUT 1 TO C; TALLY+TALLY+1);                    00020400
TALLY+0; SCAN+TALLY; GO OUX;                                     00020500

```

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 Document 314

```

C:RESETNCR: TEMP+S1; DI+NCRN; SI+LOC TEMP; DS+WDS; 00020600
SETNUMCHR: TEMP+TALLY; SI+LOC TEMP; DI+N; DS+WDS; 00020700
D: TALLY+1; SCAN+TALLY; 00020800
OUX: END; 00020900
COMMENT THE FOLLOWING DEFINES ARE USED BY PROCEDURE SUBCCHANDLER 00021000
IN SCANNING ENTITIES ON THE CARD. 00021100
1. SCANPERCENTERROR EXPECTS TO FIND ADDITIONAL INFO ON 00021200
CONTINUATION CARDS IF A * SIGN IS ENCOUNTERED AND WILL 00021300
FINALLY BRING OUT THE NEXT ENTITY WITH X=VALUE OF SCAN= 00021400
SEARCH. IF IN READING THE CARDS AN ERROR SITUATION IS 00021500
ENCOUNTERED, PROGR. WILL BRANCH TO ERROREXIT IN 00021600
SUBCCHANDLER, OR IF NO ADDITIONAL INFO ON CARDS. 00021700
2. SCANPERCENTEXIT WILL SEARCH CONTINUATION CARDS (IF 00021800
NECESSARY) AND BRING OUT THE NEXT ENTITY WITH X=VALUE OF 00021900
SCANSEARCH. IF NO MORE INFO, PROGRAM WILL BRANCH TO 00022000
GOODEXIT; 00022100
DEFINE AMORE= DO 00022200
BEGIN FILL ACCUMAL[*] WITH OCT6060606060606060; 00022300
IF NOT SCAN(ACCLMA,NA,NCR,NCR) 00022400
THEN GO TO ERROREXIT; 00022500
IF X+SCANSEARCH(ACCUMA,NA)=0 THEN GO ERROREXIT; 00022600
IF X=18 THEN 00022700
BEGIN IF NOT RETURN THEN GO TO BADGOOD[W]; 00022800
READACARD; IF NOT LEGALCC(NCR,NCR,RETURN) 00022900
THEN GO TO ERROREXIT; 00023000
END; END UNTIL X=18;#; 00023100
DEFINE BMORE= DO 00023200
BEGIN FILL ACCUMB[*] WITH OCT6060606060606060; 00023300
IF NOT SCAN(ACCUMB,NB,NCR,NCR) 00023400
THEN GO TO ERROREXIT; 00023500
IF X+SCANSEARCH(ACCUMB,NB)=0 THEN GO ERROREXIT; 00023600
IF X=18 THEN 00023700
BEGIN IF NOT RETURN THEN GO TO BADGOOD[W]; 00023800
READACARD; IF NOT LEGALCC(NCR,NCR,RETURN) 00023900
THEN GO TO ERROREXIT; 00024000
END; END UNTIL X=18;#; 00024100
DEFINE CMORE= DO 00024200
BEGIN FILL ACCUMC[*] WITH OCT6060606060606060; 00024300
IF NOT SCAN(ACCUMC,NC,NCR,NCR) 00024400
THEN GO TO ERROREXIT; 00024500
IF X+SCANSEARCH(ACCUMC,NC)=0 THEN GO ERROREXIT; 00024600
IF X=18 THEN 00024700
BEGIN IF NOT RETURN THEN GO TO BADGOOD[W]; 00024800
READACARD; IF NOT LEGALCC(NCR,NCR,RETURN) 00024900
THEN GO TO ERROREXIT; 00025000
END; END UNTIL X=18;#; 00025100
DEFINE DMORE= DO 00025200
BEGIN FILL ACCUMD[*] WITH OCT6060606060606060; 00025300
IF NOT SCAN(ACCUMD,ND,NCR,NCR) 00025400
THEN GO TO ERROREXIT; 00025500
IF X+SCANSEARCH(ACCUMD,ND)=0 THEN GO ERROREXIT; 00025600
IF X=18 THEN 00025700
BEGIN IF NOT RETURN THEN GO TO BADGOOD[W]; 00025800
READACARD; IF NOT LEGALCC(NCR,NCR,RETURN) 00025900
THEN GO TO ERROREXIT; 00026000
END; 00026100
END UNTIL X=18;#; 00026200
DEFINE ASCANPERCENTERROR= W+1; AMORE #; 00026300
DEFINE BSCANPERCENTERROR= W+1; BMORE #; 00026400
DEFINE CSCANPERCENTERROR= W+1; CMORE #; 00026500

```

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.

1	DEFINE DSCANPERCENTERROR = W+1; DMORE#;	00026600
2	DEFINE ASCANPERCENTEXIT = W+2; AMORE#;	00026700
3	DEFINE BSCANPERCENTEXIT = W+2; BMORE#;	00026800
4	DEFINE CSCANPERCENTEXIT = W+2; CMORE #;	00026900
5	DEFINE DSCANPERCENTEXIT = W+2; DMORE #;	00027000
6	DEFINE TESTENDCARD = W+3; AMORE#;	00027100
7		COMMENT LAST DEFINE TESTS
8		FOR LEGAL END CARD
9		CONTINUATION NOT ALLOWED
10		FOR END CARDS;
11	INTEGER PROCEDURE SUBCHANDLER;	00027500
12	BEGIN INTEGER W; COMMENT USED TO INDEX SWITCH BADGOOD;	00027600
13	LABEL AD, L1, L2, L3, L4, L4A, L5, L6, L7, L8, PAT, REP, R1, ID,	00027700
14	DEL, D1, D2, IDENT,	00027800
15	L3A, PATA,	00027900
16	LA,	00028000
17	ERROREXIT, GOODEXIT, ENDEXIT, OUX;	00028100
18	SWITCH BADGOOD+ ERROREXIT, GOODEXIT, ENDEXIT;	00028200
19	READACARD;	00028300
20	IF NOT LEGALCC(NCR, NCR, RETURN) THEN GO TO ERROREXIT;	00028400
21	COMMENT NOT \$\$\$ OR \$-\$. RETURN IS SET	00028500
22	TRUE IF CONTINUATION CARDS;	00028600
23	ASCANPERCENTERROR; COMMENT TERMINATE IF NOTHING ON	00028700
24	SUBCC OR ILLEGAL CARDS. ;	00028800
25	IF X=8 THEN BEGIN IF RETURN THEN	00028900
26	GO ERROREXIT; TESTENDCARD GO ERROREXIT; END;	00029000
27	COMMENT IF WE RETURN FROM TESTENDCARD	00029100
28	THEN SUBCC HAS END PLUS GARBAGE;	00029200
29	IF X=2 THEN GO TO AD; COMMENT ADD CARD;	00029300
30	IF X=7 THEN GO TO DEL; COMMENT DELETE CARD;	00029400
31	IF X=11 OR X=25 THEN GO TO PAT; COMMENT PATCH CARD;	00029500
32	IF X=14 THEN GO TO REP; COMMENT REP CARD;	00029600
33	IF X=19 THEN GO TO ID; COMMENT CARD STARTS WITH IDENTIF;	00029700
34	GO TO ERROREXIT;	00029800
35	AD:	00029900
36	ASCANPERCENTERROR	00030000
37	IF X=19 THEN GO TO ERROREXIT ;	00030100
38	ADDTOG+TRUE;	00030200
39	ADDINDEX+4; COMMENT TENTATIVELY SET SOURCE TO CARDS.	00030300
40	MAY BE CHANGED IF WE FIND ADD=ON;	00030400
41	CSCANPERCENTEXIT	00030500
42	IF X=10 THEN GO TO L2; COMMENT X=10="ON";	00030600
43	CSCANPERCENTERROR	00030700
44	IF X=4 THEN BEGIN ADDINDEX+0; GO LA; END; COMMENT "A";	00030800
45	IF X=5 THEN BEGIN ADDINDEX+1; GO LA; END; COMMENT "B";	00030900
46	IF X=6 THEN BEGIN ADDINDEX+2; GO LA; END; COMMENT "C";	00031000
47	IF X=1 THEN BEGIN ADDINDEX+3; GO LA; END; COMMENT "OCHR";	00031100
48	IF X=26 THEN BEGIN ADDINDEX+5; GO LA; END; COMMENT SOLT;	00031150
49	GO TO ERROREXIT;	00031200
50	LA:	00031300
51	L2:	00031400
52	IF X=3 THEN GO TO L3; COMMENT X=3="AFTER";	00031500
53	L1: CSCANPERCENTERROR	00031600
54	IF X=19 THEN GO TO ERROREXIT; COMMENT X=19=IDENTIFIER;	00031700
55	ADDAFTERTOG+ TRUE;	00031800
56	L3A: BSCANPERCENTEXIT GO TO L4;	00031900
57	L3: IF X=8 THEN GO TO L4; COMMENT X=8="END";	00032000
58	ADDENDTCG+TRUE; GO TO L3A;	00032100
59	L4:	00032200
60	IF X=13 THEN GO TO L5; COMMENT X=13="RENAME";	00032300
61	BSCANPERCENTERROR	00032400
62	IF X=19 THEN GO TO ERROREXIT; COMMENT LEGAL IDENTIFIER;	00032500

	RENAMETO	GO+TRUE;	00032500
L4A:	DSCANPERCENTEXIT;		00032600
L5:	IF X#9 THEN GO TO L6;	COMMENT X=9="LIST";	00032700
	LISTOG+TRUE; GO TO L4A;		00032800
L6:	IF X#12 THEN GO TO L7;	COMMENT X=12="PUNCH";	00032900
	PUNTOG+TRUE; GO TO L4A;		00033000
L7:	IF X#15 THEN GO TO L8;	DSCANPERCENTERROR	00033100
	IF X=20 THEN BEGIN SEQATOG+TRUE; SEQINC+ND; IF SEQCTOG		00033200
	THEN GO TO ERROREXIT; GO TO L4A; END ELSE GO ERROREXIT;		00033300
L8:	IF X#16 THEN GO TO ERROREXIT;		00033400
	DSCANPERCENTERROR		00033500
	IF X=20 THEN BEGIN SEQCTOG+TRUE; SEQINC+ND; IF SEQATOG		00033600
	THEN GO TO ERROREXIT; GO TO L4A; END		00033700
	ELSE GO TO ERROREXIT;		00033800
PAT:	PATCHIDG+TRUE;		00033900
	IF X=11 THEN PATCHATOG+TRUE ELSE PATCHCTOG+TRUE;		00034000
	ASCANPERCENTERROR		00034100
	IF X#19 THEN GO TO ERROREXIT; COMMENT X=19=IDENTIFIER;		00034200
	GO TO PATA;	COMMENT TEST RENAME, LIST,	00034300
		PUNCH, SEQ, ETC;	00034400
REP:	REPLACETOG+TRUE; WITHINDEX+4;		00034500
	CSCANPERCENTERROR		00034600
	IF X#19 THEN GO TO ERROREXIT; COMMENT NOT "IDENTIFIER";		00034700
	ASCANPERCENTEXIT		00034800
	IF X#17 THEN GO TO L4; COMMENT NOT "WITH";		00034900
	ASCANPERCENTERROR		00035000
	IF X#19 THEN GO TO ERROREXIT; COMMENT NOT "IDENTIFIER";		00035100
	BSCANPERCENTERROR		00035200
	IF X#10 THEN GO TO ERROREXIT; COMMENT NOT "ON";		00035300
	BSCANPERCENTERROR		00035400
	IF X=4 THEN BEGIN WITHINDEX+0; GO TO R1; END; COMMENT "A";		00035500
	IF X=5 THEN BEGIN WITHINDEX+1; GO TO R1; END; COMMENT "B";		00035600
	IF X=6 THEN BEGIN WITHINDEX+2; GO TO R1; END; COMMENT "C";		00035700
	IF X=1 THEN BEGIN WITHINDEX+3; GO TO R1; END; COMMENT OCRD;		00035800
	IF X=26 THEN BEGIN WITHINDEX+5; GO TO R1; END; COMMENT SOLT;		00035850
	GO TO ERROREXIT;		00035900
R1:	WITHTOG+TRUE; GO TO L3A;		00036000
DEL:	DELETETOG+TRUE;		00036100
	ASCANPERCENTERROR		00036200
	IF X#19 THEN GO TO ERROREXIT; COMMENT NOT IDENTIFIER;		00036300
D1:	BSCANPERCENTEXIT		00036400
	IF X#9 THEN GO TO D2;	COMMENT NOT "LIST";	00036500
	LISTOG+TRUE; GO TO D1;		00036600
D2:	IF X#12 THEN GO TO ERROREXIT;	COMMENT NOT "PUNCH";	00036700
	PUNTOG+TRUE; GO TO D1;		00036800
ID:	IF DISTOG THEN GO TO D1		00036900
	PATA: IF PATCHTOG THEN IDENTIX+LIBINIX ELSE IDENTIX+4;		00037000
	BSCANPERCENTEXIT		00037100
	IF X#10 THEN GO TO L4; COMMENT NOT "ON";		00037200
	BSCANPERCENTERROR		00037300
	IF X=4 THEN BEGIN IDENTIX+0; GO TO L3A; END; COMMENT "A";		00037400
	IF X=5 THEN BEGIN IDENTIX+1; GO TO L3A; END; COMMENT "B";		00037500
	IF X=6 THEN BEGIN IDENTIX+2; GO TO L3A; END; COMMENT "C";		00037600
	IF X=1 THEN BEGIN IDENTIX+3; GO TO L3A; END; COMMENT OCRD;		00037700
	IF X=26 THEN BEGIN IDENTIX+5; GO TO L3A; END; COMMENT SOLT;		00037750
ERROREXIT:	SUBCCHANDLER+0; GO TO OUX;		00037800
GODEXIT:	SUBCCHANDLER+1; GO TO OUX;		00037900
ENDEXIT:	SUBCCHANDLER+2; ENDTOG+TRUE;		00038000
	CLOSE(CARD, RELEASE);		00038050
OUX:	END PROCEDURE SUBCCHANDLER;		00038100

```

SAVE ARRAY DIRECTORY[C:11, 0:55];          00038200
COMMENT FOUR 3-8Y-56 DIRECTORIES.FIRST 3  00038300
ALWAYS INPUT AND LAST DIRECTORY ALWAYS NEW 00038400
ONE BEING GENERATED.                      00038500
EACH ENTRY IN THE DIRECTORY HAS:          00038600
1 CHARACTER=N=NUMBER OF CHRS IN THE IDENTIF 00038700
N CHARACTER= THE IDENTIFIER                00038800
3 CHARACTER=SRN=STARTING RECORD NUMBER.    00038900
ALSO, FIRST WORD OF EACH OF THE DIRECTORIES 00039000
CONTAINS 1,2, OR 3 CORRESPONDING TO # OF   00039100
BLOCKS IN THIS DIRECTORY.                  00039200
1) ONLY AS MANY DIR. BLOCKS AS NEEDED     00039300
2) ENTRIES NOT SPLIT ACROSS BLOCK.        00039400
3) IF N=0 AND SRN=0 THEN DIR CONTINUED IN  00039500
NEXT BLOCK                                 00039600
4) IF N=0 AND SRN≠0 THEN THIS IS THE END OF 00039700
THE DIRECTORY;                             00039800
INTEGER GT1, GT2, GT3, GTX, GTY, OUTPUT, PPP; 00039900
LABEL MAKEFUNCTION, DISPLAYFUNCTION, OUTPUT, AA, BB, CC, DD; 00040000
BOULEAN DISPSUBCC, DIRINIT;                00040100
STREAM PROCEDURE STMSPACE (BUFF);           00040200
COMMENT THIS IS CHARACTER MODE SUBPROCEDURE 00040300
OF ROUTINE "SPACE" WHICH GENERATES SPACES ON THE 00040400
LINE PRINTER;                              00040500
BEGIN DI←BUFF; 2(60(DS←LIT" "));           00040600
END STMSPACE;                               00040700
PROCEDURE SPACER(X);                         00040800
VALUE X; INTEGER X;                         00040900
BEGIN INTEGER A;                             00041000
FOR A←1 STEP 1 UNTIL ABS(X) DO              00041100
WRITE(LINE);                                00041200
STMSPACE(PRINT[0]);                          00041300
END SPACER;                                 00041400
STREAM PROCEDURE MOVE(N, A, B); VALUE N;     00041500
COMMENT MOVES N WORDS FROM A TO B;          00041600
BEGIN SI←A; DI←B; DS←N WDS;                00041700
END MOVE;                                    00041800
BOULEAN PROCEDURE READDIRECTORY(INP);        00041900
INTEGER INP;                                00042000
BEGIN                                       00042100
INTEGER B;                                  00042200
COMMENT RETURNS FALSE IF FIRST             00042300
WORD OF DIRECTORY <1 OR >3;                00042400
INTEGER C;                                  00042500
LABEL EXIT;                                 00042600
C←INP×3; C←C+1;                             00042700
READDIRECTORY←TRUE;                         00042800
READ(FIN[INP], 56, DIRECTORY[INP×3, *]);    00042900
B←DIRECTORY[INP×3, 0];                       00043000
IF B>3 OR B<1 THEN BEGIN                   00043100
READDIRECTORY←FALSE;                       00043200
GO TO EXIT;                                 00043300
END;                                         00043400
WHILE B←B-1>0 DO                             00043500
BEGIN                                       00043600
READ(FIN[INP], 56, DIRECTORY[C, *]);        00043700
00043800
00043900
00044000
00044100

```

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 Dec. 1964

```

                                C←C+1;                                00044200
                                END;                                00044300
EXIT:  END READDIRECTORY;    00044400
STREAM PROCEDURE DIRHEADING(BUFFER);    00044500
                                COMMENT HEADING FOR DIRECTORY;    00044600
                                BEGIN DI←BUFFER;                    00044700
                                20(DS←LIT"*"); DS←17 LIT"DIRECTORY IN THE ";    00044800
                                DS←34 LIT"ORDER THAT THE ENTITIES APPEAR ON ";    00044900
                                DS←4 LIT"TAPE"; 45(DS←LIT"*");    00045000
                                END DIRHEADING;                    00045100
BOOLEAN STREAM PROCEDURE STRMPRINTDIRECTORY(INDEX,LINE);    00045200
                                COMMENT CHARACTER MODE SUBPROCEDURE    00045300
                                OF ROUTINE "PRINTDIRECTORY";    00045400
                                BEGIN                                00045500
                                LOCAL N, NA;                        00045600
                                LABEL EXIT;                        00045700
                                DI←LINE; 2(DS←60 LIT" "); TALLY←1;    00045800
                                SI←INDEX; DI←LOC NA; DS←WDS; SI←NA;    00045900
                                DI←LOC N; DI←DI+7;                00046000
                                IF SC="0" THEN BEGIN TALLY←0;    00046100
                                GO TO EXIT;                        00046200
                                END;                                00046300
                                DS←CHR;                            00046400
                                DI←LINE; DI←DI+32;                00046500
                                N(DS←CHR); DI←LOC N; DI←DI+5; DS←3 CHR;    00046600
                                NA←SI; DI←INDEX; SI←LOC NA; DS←WDS;    00046700
                                DI←LINE; DI←DI+20;                00046800
                                SI←LOC N; DS←6 DEC;                00046900
EXIT:  STRMPRINTDIRECTORY←TALLY;    00047000
                                END STRMPRINTDIRECTORY;            00047100
PROCEDURE PRINTDIRECTORY(IX);    00047200
                                VALUE IX; INTEGER IX;            00047300
                                BEGIN INTEGER DIRINDEX, A, B;    00047400
                                LABEL PRNT;                        00047500
                                B←DIRECTORY[IX,0]; A←0;            00047600
                                DIRHEADING(PRINT[0]);            00047700
                                PRINTLINE; SPACER(1); DO BEGIN    00047800
                                A←A+1;                            00047900
                                DIRINDEX←IF A≠1 THEN MKABS(DIRECTORY[IX,0])    00048000
                                ELSE MKABS (DIRECTORY[IX,1]);    00048100
PRNT:  IF STRMPRINTDIRECTORY(DIRINDEX, PRINT[0])    00048200
                                THEN BEGIN                        00048300
                                PRINTLINE;                        00048400
                                GO TO PRNT;                        00048500
                                END;                                00048600
                                IX←IX+1;                            00048700
                                END                                00048800
                                UNTIL B=A=0;                        00048900
                                END PRINTDIRECTORY;                00049000
BOOLEAN STREAM PROCEDURE LOOK (ACC,N,DIR,ROW,STRIPOS,STOPOS);    00049100
                                VALUE ROW;                        00049200
                                BEGIN COMMENT LOOK SEARCHES DIRECTORY FOR    00049300
                                ITEM. REPORTS TRUE IF ITEM NOT IN DIRECTORY;    00049400
                                LOCAL DPPOS, TEMP, LGTH;            00049500
                                LABEL LOOP, EXIT;                00049600
                                SI←DIR; ROW(SI←SI+8); DPPOS←SI;    00049700
                                DI←LOC TEMP; DS←WDS; SI←TEMP;    00049800
                                SI←SI+8;                            00049900
LOOP:  DI←LOC LGTH; DI←DI+7; DS←CHR;    00050000
                                DI←N; DI←DI+7; SI←SI-1;            00050100

```

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

```

IF SC=DC                                00050200
THEN BEGIN COMMENT THE LENGTHS ARE EQUAL; 00050300
  DI←ACC;                                00050400
  IF LGTH SC=DC                          00050500
  THEN BEGIN COMMENT FOUND IT;           00050600
    DI←STRIPOS; DS←5 LIT"0"; US←3 CHR; 00050700
  IF SC="0"                               00050800
  THEN BEGIN COMMENT WE MAY BE IN THE    00050900
    WRONG ROW;                            00051000
    SI←SI+1; DI←LOC LOOK;                 00051100
    IF 3 SC=DC                            00051200
    THEN BEGIN COMMENT WRONG ROW;        00051300
      SI←DPPUS;                           00051400
      SI←SI+8;                             00051500
      DPPUS←SI;                            00051600
      DI←LOC TEMP;                         00051700
      DS←WDS;                              00051800
      SI←TEMP;                             00051900
      END                                  00052000
    ELSE SI←SI-4;                          00052100
    END;                                   00052200
    DI←LOC LGTH; DI←DI+7; DS←CHR;          00052300
    SI←SI+LGTH;                            00052400
    DI←STIPOS; DS←5 LIT"0";               00052500
    DS←3 CHR; GO TO EXIT;                 00052600
  END;                                    00052700
  SI←SI+3;                                00052800
END                                        00052900
ELSE BEGIN COMMENT LENGTHS NOT EQUAL;    00053000
  SI←SI-1;                                00053100
  IF SC="0"                               00053200
  THEN BEGIN COMMENT MAY BE NEW ROW;     00053300
    SI←SI+1; DI←LOC LOOK;                 00053400
    IF 3 SC=DC                            00053500
    THEN BEGIN COMMENT CHANGE ROWS;      00053600
      SI←DPPUS; SI←SI+8; DPPUS←SI;       00053700
      DI←LOC TEMP; DS←WDS;               00053800
      SI←TEMP;                            00053900
      END                                  00054000
    ELSE BEGIN COMMENT NOT HERE;         00054100
      TALLY←1; LOOK←TALLY;               00054200
      GO TO EXIT;                         00054300
    END;                                   00054400
  GO TO LOOP;                              00054500
  END;                                     00054600
  SI←SI+LGTH; SI←SI+4; COMMENT POSITION TO 00054700
  NEXT IDENTIFIER;                         00054800
  END;                                     00054900
  GO TO LOOP;                              00055000
EXIT:                                     00055100
END LOOK;                                  00055200
BOOLEAN STREAM PROCEDURE GETNEXTBLOB (DIR,ROWN,ROWV, TN,IV,ACC,N, 00055300
                                     GTX,GTY); 00055400
  VALUE ROWV, IV;                          00055500
  BEGIN LABEL L,M,EXIT,P,X;                00055600
  LOCAL LENGTH,B,C,D,TEMP;                 00055700
  SI←IV;                                    00055800
  DI←N;                                    00055900
  DI←DI+7;                                  00056000
  DS←CHR;                                    00056100

```

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 Location

	SI+SI-1;	00056200
	DI+LOC LENGTH;	00056300
	DI+DI+7;	00056400
	DS+CHR;	00056500
	COMMENT IDENTIFIER LENGTH NOW	00056600
	IN "N" IN OUTSIDE WORLD AND IN	00056700
	LOCAL VARIABLE "LENGTH";	00056800
	DI+ACC; LENGTH(DS+CHR);	00056900
	DI+GTX; 5(DS+LIT"0");); DS+3 CHR; D+SI;	00057000
	IF SC="0" THEN	00057100
	BEGIN COMMENT MAYBE NEXT ROW;	00057200
	SI+SI+1;	00057300
	3(CIF SC="0" THEN BEGIN SI+SI+1; GO TO L;	00057400
	END ELSE JUMP OUT 1 TO M; L:);	00057500
	SI+DIR;	00057600
	ROWV(SI+SI + 8);	00057700
	SI+ SI + 8;	00057800
	DI+LOC TEMP;	00057900
	DS+ WDS;	00058000
	TALLY+ROWV;	00058100
	TALLY+TALLY + 1;	00058200
	ROWV+TALLY;	00058300
	SI+ LOC ROWV;	00058400
	DI+ ROWN;	00058500
	DS+ WDS;	00058600
	SI+TEMP; D+SI;	00058700
X:	DI+LOC C;	00058800
	DI+DI+7;	00058900
	DS+CHR;	00059000
	C(SI+SI+1);	00059100
	DI+GTY; 5(DS+LIT"0"););DS+3 CHR;	00059200
	GO TO P; COMMENT SAVE "IN" FROM "D";	00059300
M:	COMMENT THIS WAS LAST ENTRY IN DIRECTORY;	00059400
	SI+D; SI+SI+1;	00059500
	DI+GTY; 5(DS+LIT"0");); DS+3 CHR;	00059600
	TALLY+0; GETNEXTBLOB+TALLY; GO TO EXIT;	00059700
	END; COMMENT NOW GET GTY;	00059800
	GO TO X;	00059900
P:	SI+LOC D; DI+TN; DS+WDS;	00060000
TALLY + 1; GETNEXTBLOB+ TALLY;		00060100
EXIT;	END GETNEXTBLOB;	00060200
STREAM PROCEDURE SETDIRECTION(DIR,FIL); VALUE DIR;		00060300
BEGIN LABEL LX;		00060400
LX:	SI+FIL; SKIP 2 SB; IF SB THEN BEGIN END	00060500
	ELSE GO TO LX;	00060600
	SI+ LOC DIR;	00060700
	SKIP SB;	00060800
DI+FIL;		00060900
SKIP 8 DB;		00061000
	IF SB THEN BEGIN	00061100
	DS+9 RESET; DS+SET; SKIP 4 DB;	00061200
	DS+SET;	00061300
	END	00061400
	ELSE BEGIN	00061500
	DS+4 RESET; DS+3 SET; DS+3 RESET;	00061600
	SKIP 4 DB; DS+RESET;	00061700
	END;	00061800
	END SETDIRECTION;	00061900
PROCEDURE POSTAPE (INP);		00062000
VALUE INP; INTEGER INP;		00062100
BEGIN DIRECTORY(INP*3,0)+ REC(INP)=1;		

Data Documents, 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

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

```

REC(INP)+GTX;
GT1+(GTX+(DIRECTORY(INP*3,0)+3)/5)*5 +1;
GT2+(GTX+(REC(INP)-3)/5) * 5 + 1;
GT3+(GT2-GT1)DIV 5;
SPACE(FIN(INP), GT3);
  READ(FIN(INP),56,FINARR[*1]);

```

00062200
00062300
00062400
00062500
00062600
00062700

```

  END POSTAPE;
STREAM PROCEDURE BLOBHDG(ACC,N,BUFF,STARS);
  VALUE STARS,N;
  BEGIN LOCAL X;
  LABEL A,B,C;
  TALLY+0; X+TALLY;
  SI+LOC STARS;
  DI+LOC X;
  IF B SC=DC THEN GO TO A ELSE GO TO B;
  A: DI+BUFF; DS+4 LIT "$$$ ";
  SI+ACC; GO TO C;
  B: SI+ACC; DI+BUFF;
  C: DS+N CHR;
  STARS(DS+LIT "*" );
  END BLOBHDG;

```

00062800
00062900
00063000
00063100
00063200
00063300
00063400
00063500
00063600
00063700
00063800
00063900
00064000
00064100
00064200
00064300
00064400
00064500

```

STREAM PROCEDURE GETREL (REL,BUFF);
  BEGIN
  DI+BUFF; SI+REL;
  DS+4 DEC;
  END;
PROCEDURE PRINTPUNCH(P,IMG,REL);
  VALUE P,IMG,REL;
  INTEGER P,IMG,REL;
  BEGIN
  IF P#2 THEN
  BEGIN MOVE(10,FINARR[IMG*11+1], PRINT[2]);
  GETREL(REL,PRINT[13]);
  PRINTLINE;
  STMSPACE(PRINT[0]);
  END;
  IF P>1 THEN
  BEGIN MOVE(10,FINARR[IMG*11+1], PUNCH(0));
  RELEASE(PUNCH);
  END;
  END PRINTPUNCH;

```

00064600
00064700
00064800
00064900
00065000
00065100

```

STREAM PROCEDURE SBLP (BUFF);
  BEGIN
  DI+BUFF; 2(40(DS+LIT " "));
  END SBLP;
PROCEDURE BLANKPUNCH;
  BEGIN SBLP(PUNCH(0));
  END BLANKPUNCH;
PROCEDURE PRINTPUNALL(P);
  VALUE P;
  INTEGER P; BEGIN LABEL A;
  LABEL B;
  BOOLEAN BB;
  BB+GETNEXTBLOB(DIRECTORY, ROW,ROW,II,II,
  ACCUMA,NA,GTX,GTY);
  POSTAPE(LIBINIX);
  IF P#2 THEN BEGIN

```

00065200
00065300
00065400
00065500
00065600
00065700
00065800
00065900
00066000
00066100
00066200
00066300
00066400
00066500
00066600
00066700
00066800
00066900
00067000
00067100
00067200
00067300
00067400
00067500
00067600
00067700
00067800
00067900
00068000
00068100

```

1      PAG;                                00068200
2      BLOBHDG(ACCUMA, NA, PRINT[0], 120=NA); 00068300
3      PRINTLINE;                          00068400
4      SPACER(1);                          00068500
5      END;                                00068600
6      IF P>1 THEN BEGIN                   00068700
7      BLANKPUNCH;                         00068800
8      BLOBHDG(ACCUMA, NA, PUNCH(0), 0);   00068900
9      RELEASE(PUNCH);                     00069000
10     END;                                00069100
11     REL+0;                              00069200
12     IMG+(REC[LIBINIX]-1)MOD 5;          00069300
13     REL+REL + 1;                        00069400
14
15     WHILE REC[LIBINIX]#GTY DO BEGIN      00069500
16     PRINTPUNCH(P, IMG, REL);            00069600
17     GTY+GTY;                            00069700
18     REC[LIBINIX] ← REC[LIBINIX] + 1;    00069800
19     IMG+IMG+1;                          00069900
20     REL+REL + 1;                        00070000
21     IF IMG>4 THEN BEGIN IMG+0;         00070100
22     READ(FIN[LIBINIX], 56, FINARR[*]);  00070200
23     END; END;                           00070300
24
25     IF B3 THEN GO TO A;                  00070400
26     END PRINTPUNALL;                    00070500
27
28     BOOLEAN PROCEDURE SCANMASTER;       00070600
29     BEGIN LABEL ERROREXIT, GOODEXIT, MAKE, DISPLAY, LISPUN; 00070700
30     LABEL A;                             00070800
31     INTEGER W;                            00070900
32     SWITCH BADGOOD ← ERROREXIT, GOODEXIT; 00071000
33     BOOLEAN B;                            00071100
34     PRINTCC(PRINT[0], CARD(0)); PRINTLINE; 00071200
35     SCANMASTER+TRUE;                     00071300
36     SPACER(1);                           00071400
37     ZOT(ZOTTEMP, "%", CARD(9));          00071500
38     NCR+MKABS(CARD(0));                   00071600
39     B+LEGALCC(NCR, NCR, RETURN);         00071700
40     IF B AND NOT RETURN THEN BEGIN END   00071800
41     ELSE GO TO ERROREXIT;                00071900
42     ASCANPERCENTERROR;                   00072000
43     IF X=24 THEN GO TO DISPLAY;          00072100
44     IF X=21 THEN GO TO MAKE; GO TO ERROREXIT; 00072200
45     MAKE: ASCANPERCENTERROR;             00072300
46     IF X>3 AND X<7 THEN LIBOUTIX←X-4 ELSE 00072400
47     GO TO ERROREXIT;                     00072500
48     MKTOG+TRUE;                          00072600
49     ASCANPERCENTEXIT;                    00072700
50     IF X=22 THEN BEGIN MKTOG+FALSE; MKFTOG+TRUE; 00072800
51     ASCANPERCENTERROR;                   00072900
52     IF X>3 AND X<7 THEN LIBINIX←X-4 ELSE 00073000
53     GO TO ERROREXIT; GO TO LISPUN; END; 00073100
54     GO TO A;                             00073200
55     DISPLAY: DISTOG+TRUE;                 00073300
56     ASCANPERCENTERROR;                   00073400
57     IF X>3 AND X<7 THEN LIBINIX←X-4 ELSE 00073500
58     GO TO ERROREXIT;                     00073600
59     LISPUN: ASCANPERCENTEXIT;             00073700
60
61     GO TO ERROREXIT;                     00073800
62
63     IF X>3 AND X<7 THEN LIBINIX←X-4 ELSE 00073900
64     GO TO ERROREXIT;                     00074000
65
66     LISPUN: ASCANPERCENTEXIT;             00074100

```

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

A:	IF X=9 THEN MLISTOG+TRUE	00074200
	ELSE IF X=20 THEN COPYTUG+TRUE	00074300
	ELSE IF X=12 THEN MPUNIOG+TRUE	00074400
	ELSE IF X=23 THEN DIRTUG+TRUE	00074500
	ELSE IF X=25 THEN SING+TRUE	00074600
	ELSE GO TO ERROREXIT;	00074700
	ASCANPERCENTEXIT;	00074800
	GO TO A;	00074900
	ERROREXIT: SCANMASTER+FALSE;	00075000
	GOODEXIT: END SCANMASTER;	00075100
	STREAM PROCEDURE FOOLROY(FIEL,CONTROL,SIZE); VALUE CONTROL,SIZE;	00075200
	BEGIN COMMENT FOOLROY WILL SET THE CONTROL FIELD OF THE	00075300
	I/O DESCRIPTOR FIEL TO CONTROL AND THE SIZE TO SIZE;	00075400
	DI=FIEL;	00075500
	DI=DI+2;	00075600
	SI=LOC SIZE;	00075700
	SI=SI+7;	00075800
	DS=CHR;	00075900
	SI=LOC CONTROL;	00076000
	SI=SI+6;	00076100
	DS=2 CHR;	00076200
	END FOOLROY;	00076300
	BOOLEAN STREAM PROCEDURE ENTERDIRECTORY (DIR, ROWN, ROWV, POSN, POSV,	00076400
	ACC, N, RC);	00076500
	COMMENT WILL ENTER THIS ENTITY IN DIRECTORY	00076600
	IF N, ACC AND RC WILL STILL LEAVE 4 CHARS.	00076700
	AT THE END OF A DIRECTORY BLOCK, PROCEDURE	00076800
	WILL AUTOMATICALLY UPDATE PARAMETERS SHOWING	00076900
	LAST ROW AND NEXT CHARACTER POSITION TO BE	00077000
	TESTED FOR POSSIBLE ENTRY. PROCEDURE RETURNS	00077100
	TRUE IF ENTITY ENTERED IN DIRECTORY	00077200
	FALSE IF ENTITY NOT ENTERED INDICATING	00077300
	THERE WOULD HAVE BEEN DIRECTORY	00077400
	OVERFLOW;	00077500
	VALUE ROWV, POSV, N;	00077600
	BEGIN LABEL TRYNEXTROW, FIT, NOTFIT, ENTER, EXIT;	00077700
	LOCAL LASTROW, ENDOFROW, TEMP, X, Y;	00077800
	TALLY+11; COMMENT ELEVEN IS LAST ROW OF	00077900
	ARRAY DIRECTORY;	00078000
	LASTROW+TALLY;	00078100
	SI=DIR;	00078200
	ROWV(SI+SI+8);	00078300
	DI=LOC TEMP;	00078400
	DS+WDS; SI+TEMP;	00078500
	56(SI+SI+8); COMMENT POINT END OF BLK;	00078600
	SI=SI-4; COMMENT BACK UP 4 CHARS;	00078700
	ENDOFROW+SI; COMMENT WILL BE ENDOFROW.	00078800
	FIRST TEST WILL BE FOR DUMMY	00078900
	ENTRY WHERE =0 AND NEWRC	00079000
	CONTAINS ENDING REQRD # OF THE	00079100
	PREVIOUS ENTITY WHICH WAS	00079200
	ENTERED IN THE DIRECTORY;	00079300
	DI=POSV; COMMENT POINT INTO THE DIRECTORY	00079400
	BLOCK ITSELF;	00079500
	SI=LOC N; SI=SI+7;	00079600
	IF SC="0" THEN BEGIN DS=CHR; SI=RC;	00079700
	SI=SI+3; DS=3 CHR;	00079800
	GO TO FIT;	00079900
	END;	00080000
	SI=LOC ENDOFROW;	00080100

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.

```

4(X+DI; DI+LOC X; IF 8 SC=DC THEN JUMP OUT 1 TO TRYNEXTROW; SI+SI-8; 00080200
DI+X; DI+DI+1;); 00080300
N(X+DI; DI+LOC X; IF 8 SC=DC THEN JUMP OUT 1 TO TRYNEXTROW; SI+SI-8; 00080400
DI+X; DI+DI+1;); 00080500
ENTER: DI+POSV; 00080600
SI+LOC N; SI+SI+7; 00080700
DS+CHR; 00080800
SI+ACC; N(DS+CHR); 00080900
SI+RC; SI+SI+5; 00081000
DS+3 CHR; 00081100
X+DI; SI+LOC X; DI+POSN; DS+WDS; 00081200
GO TO FIT; 00081300
TRYNEXTROW: DI+POSV; 4(DS+LIT "0"); 00081400
SI+LOC ROWV; 00081500
DI+LOC LASTROW; 00081600
IF 8 SC=DC THEN GO TO NOTFIT; 00081700
TALLY+ROWV; TALLY+TALLY+1; 00081800
ROWV+TALLY; 00081900
SI+LOC ROWV; DI+ROWN; DS+WDS; 00082000
SI+DIR; ROWV(SI+SI+8); DI+LOC X; DS+WDS; 00082100
SI+DIR; COMMENT UPDATE # DIRECTORY 00082200
BLOCKS IN WORD ONE OF 00082300
DIRECTORY BLOCK #1; 00082400
9(SI+SI+8); 00082500
DI+LOC TEMP; 00082600
DS+WDS; 00082700
SI+TEMP; SI+SI+7; DI+LOC Y; DI+DI+7; DS+CHR; 00082800
TALLY+Y; TALLY+TALLY+1; Y+TALLY; 00082900
DI+TEMP; SI+LOC Y; DS+WDS; 00083000
DI+X; GO TO ENTER; 00083100
FIT: TALLY+1; ENTERDIRECTORY+TALLY; GO TO EXIT; 00083200
NOTFIT: TALLY+0; ENTERDIRECTORY+TALLY; 00083300
EXIT: END ENTERDIRECTORY; 00083400
BOOLEAN STREAM PROCEDURE DATACARD(BUFF); 00083500
BEGIN LOCAL X; 00083600
LABEL A,B,C; 00083700
DI+LOC X; DI+DI+2; 00083800
DS+ 6 LIT "ssss-s"; DI+DI-6; 00083900
SI+BUFF; 00084000
IF 3 SC=DC THEN GO TO B; SI+SI-3; 00084100
IF 3 SC=DC THEN GO TO B ELSE GO TO A; 00084200
A: TALLY+1; GO TO C; 00084300
B: TALLY+0; 00084400
C: DATACARD+TALLY; 00084500
END DATACARD; 00084600
STREAM PROCEDURE GETSEQ(DUM,N,SEK); 00084700
VALUE N; 00084800
BEGIN 00084900
SI+DUM; DI+SEK; 00085000
DS+N CHR; 00085100
END GETSEQ; 00085200
STREAM PROCEDURE SETSEQ (DUM,N,SEK); 00085300
VALUE N; 00085400
BEGIN 00085500
SI+SEK; DI+DUM; 00085600
DS+N DEC; 00085700
END SETSEQ; 00085800
INTEGER STREAM PROCEDURE COMPARE(CAR,TAP,N); 00085900
VALUE CAR, TAP, N; BEGIN LABEL A; 00086000
SI+LOC CAR; DI+LOC TAP; 00086100

```

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

```

IF N SC>DC THEN BEGIN TALLY+0; GO TO A; END; 00086200
SI+SI-N; DI+DI-N; 00086300
IF N SC=DC THEN BEGIN TALLY+1; GO TO A; END 00086400
ELSE TALLY+2; 00086500
A: COMPARE+TALLY; 00086600
END COMPARE; 00086700
INTEGER PROCEDURE TAPEMAKER(INP,ACC,N); 00086800
VALUE INP; 00086900
INTEGER INP,N; 00087000
ARRAY ACC(0); 00087100
BEGIN 00087200
BOCLEAN PATCHING, LISTING, PUNCHING, 00087300
BOOL, 00087400
PATCHCARDIN, NEEDPATCHCARD, RENUMING; 00087500
INTEGER MASTERSEQ, CARDSEQ, LASTRESEQ; 00087600
LABEL A,B,C,D,E,F,G,H,J,L,M,EXIT; 00087700
COMMENT SET SOME INITIAL 00087800
PARAMETERS*****; 00087900
IMGOBT+(NEWRC-1)MOD 5; 00088000
REL+1; 00088100
IF SEQATOG OR SEQCTOG THEN RENUMING+TRUE; 00088200
IF PATCHATOG OR PATCHCTOG THEN PATCHING+TRUE; 00088300
IF PPP=1 OR PPP=3 THEN LISTING+TRUE; 00088400
IF PPP>1 THEN PUNCHING+TRUE; 00088500
IF LISTING THEN BEGIN 00088600
PAG; BLOBHDG(ACC,N, PRINT(0),120-N); 00088700
PRINTLINE; SPACER(1); 00088800
END; 00088900
IF PUNCHING THEN BEGIN 00089000
BLANKPUNCH; BLOBHDG(ACC,N,PUNCH(0), 0); 00089100
RELEASE (PUNCH); 00089200
END; 00089300
IF INP<3 THEN IMG+(RECLINP-1)MOD 5; 00089400
GO TO IF INP<3 THEN A 00089500
ELSE IF INP=3 THEN B 00089600
ELSE IF INP=5 THEN B 00089650
ELSE C; 00089700
COMMENT A,B,C 00089800
TAPE IS INPUT; A: IF RECLINP]=GTI THEN GO TO H; COMMENT LAST IMAGE 00089900
HAS BEEN READ; 00090000
MOVE(10,FINARR[IMG*11+1],DUMMY); 00090100
D: IF NOT PATCHING THEN GO TO E; 00090200
IF PATCHATOG THEN GETSEQ(DUMMY[9],8,MASTERSEQ) 00090300
ELSE GETSEQ(DUMMY[0],6,MASTERSEQ); 00090400
IF PATCHCARDIN THEN GO TO G ELSE GO TO F; 00090500
COMMENT OCRDIMG 00090600
TAPE IS INPUT; B: READ(FIN[INP],10,DUMMY[*])[H]; GO TO D; 00090700
H: IF INP=3 OR INP=5 THEN LOCK(FIN[INP],RELEASE); 00090800
IF PATCHING THEN GO TO J ELSE GO TO EXIT; 00090850
COMMENT CARD 00090900
DECK IS INPUT- 00091000
MAY BE PATCH 00091100
CARDS SO TEST; C: IF NOT PATCHING THEN BEGIN 00091200
M: RELEASE(CARD); 00091300
MOVE(10,CARD(0),DUMMY); 00091400
IF DATACARD(CARD(0)) THEN GO TO E 00091500
ELSE BEGIN CCBUFF+TRUE; GO EXIT; END; 00091600
END; 00091700
COMMENT THIS IS 00091800
A PATCH CARD; F: RELEASE(CARD); MOVE(10, CARD(0),PATCHDUMMY); 00091900

```

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

```

IF DATACARD(CARD(0))                                00092000
THEN BEGIN                                           00092100
IF PATCHATOG THEN GETSEQ(PATCHDUMMY(9),8,CARDSEQ) 00092200
ELSE GETSEQ(PATCHDUMMY(0),6,CARDSEQ);              00092300
PATCHCARDIN+TRUE;                                  00092400
GO TO G;                                             00092500
END                                                  00092600
ELSE BEGIN                                           00092700
CCBUFF+TRUE;PATCHING+FALSE;                       00092800
NEEDPATCHCARD+FALSE; GO TO E;                    00092900
END;                                                 00093000
J: PATCHING+FALSE; COMMENT ENTERED WHEN A,B,C OR 00093100
UCRDIMG IS COMPLETE BUT PATCH                      00093200
CARDS MAY STILL HAVE TO BE ADDED.00093300
PATCHING IS SET FALSE BUT CARDS                 00093400
ARE READ AND TAGGED ON TO THE END00093500
OF THE PREVIOUS TAPE IMAGES;                      00093600
NEEDPATCHCARD+TRUE;                               00093700
IF PATCHCARDIN THEN GO TO L ELSE GO TO M;          00093800
COMMENT TEST                                        00093900
WHICH TO TAKE=                                     00094000
PATCH CARD OR                                     00094100
TAPE IMAGE. IF                                     00094200
CARD IS LESS IT                                    00094300
GOES. IF CARD IS                                   00094400
EQUAL IT GOES AND                                  00094500
REPLACES TAPE IMAGE=                               00094600
IF CARD IS GREATER                                 00094700
IT WAITS;                                          00094800
G: X+ COMPARE(CARDSEQ,MASTERSEQ, IF PATCHATOG THEN 8 ELSE 6); 00094900
IF X=0 THEN BEGIN                                  00095000
NEEDPATCHCARD+FALSE; GO TO E; END;               00095100
IF X=1 THEN BEGIN                                  00095200
NEEDPATCHCARD+FALSE;GO TO L; END;                00095300
NEEDPATCHCARD+TRUE; COMMENT CARD LESS THAN      00095400
TAPE IMAGE;                                        00095500
BOCL+TRUE; MOVE(10,DUMMY,ACCUME);                 00095600
L: PATCHCARDIN+FALSE; MOVE(10,PATCHDUMMY,DUMMY); 00095700
COMMENT TEST FOR RESEQUENCING; E: IF RENUMING THEN BEGIN 00095800
LASTRESEQ+LASTRESEQ + SEQINC;                    00096000
IF SEQATOG THEN SETSEQ(DUMMY(9),8,LASTRESEQ) 00096100
ELSE SETSEQ(DUMMY(0),6,LASTRESEQ);               00096200
END;                                               00096300
COMMENT NOW MAKE NEW TAPE;                         00096400
IF IMGOUT=0 THEN MOVE(1,NEWRC,FOUT[LIBOUTIX](0)) 00096500
MOVE(10,DUMMY,FOUT[LIBOUTIX](IMGOUT*11 +1));    00096600
IF LISTING THEN BEGIN                              00096700
MOVE(10,FOUT[LIBOUTIX](IMGOUT*11+1),PRINT[2]); 00096800
GETREL(REL,PRINT[13]);                             00096900
PRINTLINE;                                         00097000
END;                                               00097100
IF PUNCHING THEN BEGIN                             00097200
MOVE(10, FOUT[LIBOUTIX](IMGOUT*11+1),PUNCH(0)); 00097300
RELEASE(PUNCH);                                    00097400
END;                                               00097500
COMMENT RESET PARAMETERS FOR ANOTHER PASS=DUMP 00097600
OUTPUT TAPE BLOCK                                 00097700

```

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

Date documents, inc.

```

IF NECESSARY;          REL+REL+1;          00098000
IF BOOL THEN BEGIN MOVE(10,ACCUM, DUMMY); 00098100
BOOL+FALSE; END;      00098200
IF NEEDPATCHCARD THEN BEGIN END          00098300
ELSE IF INP<3 THEN BEGIN                   00098400
IF RECC[INP]#GTIY THEN BEGIN               00098500
REC[INP]+REC[INP] + 1;                     00098600
IMG+IMG+1; IF IMG>4 THEN BEGIN             00098700
IMG+0; READ(FIN[INP],56,FINARR[*]);;END;END;END;00098800
NEWRC+NEWRC+1;                             00098900
IMGOUT+IMGOUT+1;                           00099000
IF IMGOUT=5 THEN BEGIN                     00099100
WRITE(FOUT[LIBOUTIX]);                     00099200
IMGOUT+0; END;                             00099300
GO 10.                                     00099400
IF NEEDPATCHCARD THEN C                   00099500
ELSE IF INP<3 THEN A                       00099600
ELSE IF INP=3 THEN B                       00099700
ELSE IF INP=5 THEN B                       00099750
ELSE C;                                    00099800
EXIT: IF LISTING THEN PAGE; TAPEMAKER+0; 00099900
END TAPEMAKER;                             00100000
STREAM PROCEDURE ADDR(FIEL,SAVED);          00100100
BEGIN SI+FIEL; SI+SI+5;                    00100200
DI+SAVED; DI+DI+5;                         00100300
DS+3 CHR;                                  00100400
END ADDR;                                  00100500
BOOLEAN PROCEDURE OPENED (INPUT);          00100600
VALUE INPUT; INTEGER INPUT;               00100700
BEGIN LABEL EXIT;                          00100800
IF OPEN[INPUT] THEN BEGIN OPENED+TRUE; GO EXIT; 00100900
END;                                        00101000
IF NOT READDIRECTORY(INPUT) THEN BEGIN     00101100
OPENED+FALSE; GO EXIT; END;               00101200
OPENED+TRUE; OPEN[INPUT]+TRUE; REC[INPUT]+2; 00101300
EXIT; END OPENED;                          00101400
INTEGER PROCEDURE MOVEUP(STOP, ALL);        00101500
COMMENT MOVEUP TRANSFERS PROGRAMS FROM    00101600
INPUT MASTER TO NEW OUTPUT TAPE          00101700
UNTIL STOP IS REACHED WHERE STOP         00101800
= 1ST CARD IMAGE OF PROGRAM NOT          00101900
XFERED.                                   00102000
MOVEUP=0=OK                               00102100
MOVEUP=1=DIRECTORY OVERFLOW               00102200
MOVEUP=2=NOT FORWARD ON TAPE;            00102300
VALUE ALL,STOP;                            00102400
INTEGER STOP, ALL;                         00102500
BEGIN LABEL A;                             00102600
INTEGER J,K ;                              00102700
BOOLEAN SA,SC,PA,PC;                       00102800
SA+SEQATOG;                                00102900
SC+SEQCTOG;                                00103000
PA+PATCHATOG;                             00103100
PC+ PATCHCTOG;                             00103200
SEQATOG+SEQCTOG+PATCHATOG+PATCHCTOG+FALSE; 00103300
IF ALL=1 THEN                              00103400
BEGIN WHILE B1 DO                          00103500
BEGIN B1+GETNEXTBLOB(DIRECTORY,ROW,ROW,II,II,ACCUMD,ND, 00103600
GTX,GTY);                                  00103700
IF ENTERDIRECTORY(DIRECTORY,EDROW,EDROW,EDPOS,EDPOS, 00103800

```



```

OUTPUTV+5; GO OUTPUT; END; 00109800
RECL(LIBINIX)+-2; COMMENT SET INPUT TAPE POSITION; 00109900
ROW+LIBINIX * 3; 00110000
II+ MKABS(DIRECTORY(LIBINIX * 3, 1)); 00110100
COMMENT INITIALIZE PARAMETERS IN CASE 00110200
PROCEDURE GETNEXTBLOB IS CALLED TO 00110300
LOOK AT THE INPUT TAPE; 00110400
B1+ TRUE; 00110500
END; 00110600
IF DISTOG AND DIRTOG THEN BEGIN 00110700
PRINTDIRECTORY(LIBINIX*3); PAG; END; 00110800
IF NOT DISTOG THEN BEGIN 00110900
BCOUT+1; EDROW+9; EDPOS+MKABS(DIRECTORY[9,1]); 00111000
DIRECTORY[9,0]+1; 00111100
WRITE(FOUT(LIBOUTIX)); 00111200
WRITE(FOUT(LIBOUTIX)); 00111300
WRITE(FOUT(LIBOUTIX)); 00111400
00111500
00111600
00111700
00111800
00111900
00112000
00112100
END; 00112200
LISTOG+PUNTOG+WITHTOG+AFTERTOG+SEQATOG+SEQCTOG+ 00112300
ADDTOG+ADDENDTOG+ADDAFTERTOG+PATCHTOG+REPLACETOG+ 00112400
PATCHATOG+PATCHCTOG+ 00112500
DELETETOG+RENAMETOG+FALSE; PPP+0; 00112600
TEMP+ SUBCCHANDLER ; 00112700
IF TEMP=0 THEN BEGIN 00112800
OUTPUTV+2; GO OUTPUT; END; 00112900
COMMENT ERROR IN SUBCC; 00113000
COMMENT TEMP=0=ERROR 00113100
TEMP=1=SUBCC 00113200
TEMP=2=END CARD; 00113300
IF NOT DISTOG THEN GO TO MAKEFUNCTION; 00113400
IF TEMP=2 THEN 00113500
BEGIN IF DISPSUBCC THEN GO TO BB; 00113600
IF MLISTOG THEN PPP.[47:1]+1; 00113700
IF MPUNTOG THEN PPP.[46:1]+1; 00113800
IF MLISTOG OR MPUNTOG THEN BEGIN 00113900
PRINTPUNALL(PPP); 00114000
OUTPUTV+3; 00114100
GO OUTPUT; 00114200
END; 00114300
IF DIRTOG THEN GO TO BB; 00114400
PRINTDIRECTORY(LIBINIX*3); GO TO BB; 00114500
END; DISPSUBCC+TRUE; 00114600
IF NOT MLISTOG AND NOT MPUNTOG AND NOT LISTOG 00114700
AND NOT PUNTOG THEN 00114800
GO TO MORE; 00114900
IF LISTOG OR MLISTOG THEN PPP.[47:1]+1; 00115000
IF PUNTOG OR MPUNTOG THEN PPP.[46:1]+1; 00115100
PAG; 00115200
IF LOOK(ACCUMA,NA,DIRECTORY,LIBINIX*3,GTX,GTY) 00115300
THEN BEGIN OUTPUTV+4; 00115400
GO OUTPUT; END; 00115500
COMMENT ITEM NOT IN DIRECTORY; 00115600
POSTAPE(LIBINIX); 00115700

```

MORE:

DISPLAYFUNCTION:

```

COMMENT AFTER POSTAPE GTX OR STARTING 00115800
RECORD IN RC; 00115900
IF PPP=2 THEN BEGIN COMMENT LISTING REQUIRED; 00116000
    BLOBHDG(ACCUMA, NA, PRINT(0), 120=NA); 00116100
PRINTLINE; 00116200
SPACER(1); COMMENT ALSO BLANKS PRINT BUFF; 00116300
    END; 00116400
IF PPP>1 THEN BEGIN 00116500
    BLANKPUNCH; 00116600
    BLOBHDG(ACCUMA, NA, PUNCH(0), 0); 00116700
RELEASE(PUNCH); 00116800
    END; 00116900
IMG←(REC[LIBINIX]-1)MOD 5; 00117000
REL←1; 00117100
WHILE REC[LIBINIX]≠GTY DU BEGIN 00117200
    PRINTPUNCH(PPP, IMG, REL); 00117300
    IF REC[LIBINIX]≠GTY THEN BEGIN 00117500
REC[LIBINIX] ← REC[LIBINIX] + 1 ; 00117600
        IMG←IMG+1; REL←REL+1; 00117700
        IF IMG>4 THEN BEGIN IMG←0; 00117800
READ(FIN[LIBINIX], 56, FINARR[*]); 00117900
            END; END; END; 00118000
        PAG; 00118100
        GO TO MORE; 00118200
    BB: OUTPUTV←3; GO TO OUTPUT; 00118300
MAKEFUNCTION: BEGIN LABEL DELETETYPE, REPLACETYPE, PATCHTYPE, ADTOTYPE 00118400
    , AA2 , EOF , IDENTONATYPE, WRAPUP; 00118500
    LABEL A2, ADDONE; 00118600
    INTEGER A, B, IMGOUT, TESTMAKER; 00118700
    IF DELETETOG THEN GO TO DELETETYPE ELSE 00118800
    IF REPLACETOG THEN GO TO REPLACETYPE ELSE 00118900
    IF PATCHTOG THEN GO TO PATCHTYPE ELSE 00119000
    IF ADTOTYPEG THEN GO TO ADTOTYPE ELSE 00119100
    IF ENDTOG THEN GO TO WRAPUP ELSE 00119200
    GO TO IDENTONATYPE; 00119300
IDENTONATYPE: IF LISTOG OR MLISTOG THEN PPP.[47:1]+1; 00119400
    IF PUNTOG OR MPUNTOG THEN PPP.[46:1]+1; 00119500
    IF IDENTIX≤3 THEN BEGIN 00119600
    IF MKFTOG THEN BEGIN IF IDENTIX=LIBINIX THEN 00119700
    BEGIN OUTPUTV+2; GO OUTPUT; END; END; 00119800
    IF OPENED (IDENTIX) THEN BEGIN END ELSE BEGIN 00119900
        OUTPUTV+5; GO OUTPUT; END; 00120000
        IF LOOK(ACCUMA, NA, DIRECTORY, IDENTIX*3, GTX, GTY) 00120100
        THEN BEGIN OUTPUTV+4; GO OUTPUT; 00120200
            END; 00120300
            POSTAPE(IDENTIX); END; 00120400
            IF RENAMETOG THEN BEGIN 00120500
IF ENTERDIRECTORY(DIRECTORY, EDROW, EDROW, EDPOS, EDPOS, 00120600
ACCUMB, NB, NEWRC) THEN 00120700
            ELSE BEGIN OUTPUTV+10; GO OUTPUT; END; END 00120800
            ELSE BEGIN 00120900
IF ENTERDIRECTORY(DIRECTORY, EDROW, EDROW, EDPOS, EDPOS, 00121000
ACCUMA, NA, NEWRC) THEN 00121100
            ELSE BEGIN OUTPUTV+10; GO OUTPUT; END; 00121200
            END; 00121300
            IF RENAMETOG THEN TEMP←TAPEMAKER(IDENTIX, ACCUMB, NB) 00121400
            ELSE TEMP←TAPEMAKER(IDENTIX, ACCUMA, NA); 00121500
            GO TO MORE; 00121600
DELETETYPE: BEGIN LABEL U4, DELETEDONE; 00121700

```

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

Date Document/line

	IF MKTUG THEN GO TO E7 ELSE	00121800
	IF LOOK(ACCUMA,NA,DIRECTORY,LIBINIX * 3,GTX,GTY) THEN	00121900
	GO TO E8; PFRQMM;	00122000
	GTS←MOVEUP(GTX,0);	00122100
	TEST1;	00122200
	B1←GETNEXTBLOB(DIRECTORY,ROW,ROW,II,II,ACCUMD,ND,	00122300
	GTX,GTY);	00122400
	PPP←0; REL←1; IMGOUT←(REC[LIBINIX]-1) MOD 5;	00122500
	IF PUNTUG THEN BEGIN PPP,[46;1]←1;	00122600
	BLOBHDG(ACCUMD,ND,PUNCH(0),0); RELEASE(PUNCH); END;	00122700
	IF LISTUG THEN BEGIN PPP,[47;1]←1; PAG;	00122800
	BLOBHDG(ACCUMD,ND,PRINT[0],120-ND); PRINTLINE;	00122900
	SPACER(2); END;	00123000
	IF PPP=0 THEN BEGIN GTX←GTY; POSTAPE(LIBINIX);	00123100
	GO TO DELETEDONE; END; POSTAPE(LIBINIX);	00123200
D4:	IF REC[LIBINIX]≥GTY THEN GO TO DELETEDONE;	00123300
	PRINTPUNCH(PPP,IMGOUT,REL); IMGOUT←IMGOUT+1;	00123400
	REC[LIBINIX]←REC[LIBINIX]+1; REL←REL+1;	00123500
		00123600
	IF IMGOUT ≠ 5 THEN GO TO D4;	00123700
	IMGOUT←0; READ(FIN[LIBINIX],50,FINARR[*]); GO D4;	00123800
	DELETEDONE: GO TO MORE; END DELETEDONE;	00123900
	PATCHTYPE: BEGIN LABEL P1, P2, P3;	00124000
	INTEGER J;	00124100
	IF (PATCHATUG AND SEGCTUG)	00124200
	OR (PATCHCTUG AND SEGATUG) THEN GO TO E2;	00124300
	IF MKTUG THEN	00124400
	BEGIN IF IDENTIX < 3 THEN	00124500
	BEGIN P3: IF OPENED(IDENTIX) THEN BEGIN END ELSE	00124600
	GO TO E5;	00124700
	IF LOOK(ACCUMA,NA,DIRECTORY,IDENTIX * 3,GTX,GTY)	00124800
	THEN GO TO E4	00124900
	ELSE POSTAPE(IDENTIX);	00125000
	END;	00125100
	P1: IF RENAMETUG THEN	00125200
	BEGIN	00125300
	IF ENTERDIRECTORY(DIRECTORY,EDROW,EDROW,EDPOS,EDPOS,	00125400
	ACCUMB,NB,NEWRC) THEN	00125500
	ELSE GO TO E10; END ELSE	00125600
	BEGIN	00125700
	IF ENTERDIRECTORY(DIRECTORY,EDROW,EDROW,EDPOS,EDPOS,	00125800
	ACCUMA,NA,NEWRC) THEN	00125900
	ELSE GO TO E10; END; GO TO P2;	00126000
	END ELSE	00126100
	BEGIN IF IDENTIX=LIBINIX THEN	00126200
BEGIN	IF LOOK(ACCUMA,NA,DIRECTORY,IDENTIX * 3,GTX,GTY)	00126300
	THEN GO TO E4;	00126400
	PFRQMM; J←GTX; TEMP←MOVEUP(GTX,0);	00126500
	IF TEMP=1 THEN GO TO E10;	00126600
	IF TEMP=2 THEN GO TO E9; GTX←J;	00126700
	B1←GETNEXTBLOB(DIRECTORY,ROW,ROW,II,II,ACCUMA,NA,GTX,GTY);	00126800
	POSTAPE(LIBINIX); GO TO P1;	00126900
	END ELSE	00127000
	BEGIN IF IDENTIX>2 THEN GO TO P1 ELSE GO TO P3;	00127100
	END;	00127200
	END;	00127300
P2:	PFRQMB;	00127400
	IF RENAMETUG THEN TEMP←TAPEMAKER(IDENTIX,ACCUMB,NB)	00127500
	ELSE TEMP←TAPEMAKER(IDENTIX,ACCUMA,NA);	00127600
	GO TO MORE; END PATCHTYPE;	00127700

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

Date Documented: 11/1/77

REPLACETYPE: ADDBTYPE:	IF REPLACETOG THEN ADDINDEX+WITHINDEX;	00127800
	IF MKTTOG OR LIBINIX=ADDINDEX THEN GO TO E2;	00127900
	IF ADDAFIERIDG OR REPLACETOG	00128000
	THEN BEGIN IF LOOK(ACCUMC,NC,DIRECTORY,LIBINIX * 3,	00128100
	GTX, GTY) THEN GO TO E4;	00128200
	END	00128300
	ELSE IF ADDENDTOG	00128400
	THEN BEGIN IF B1 THEN BEGIN PFROMM; GT5+MOVEUP(GTY,1);	00128500
	TEST1; END; GO TO AA2;	00128600
	END ELSE GO TO AA2;	00128700
	IF ADDAFTERTOG THEN BEGIN PFROMM;	00128800
	GT5+MOVEUP(GTY,0); TEST1; END	00128900
	ELSE BEGIN PFROMM;	00129000
	GT5+MOVEUP(GTX,0); TEST1; PPP+0;	00129100
	B1+GETNEXTBLOB(DIRECTORY,ROW,ROW,II,II,	00129200
	ACCUMD, ND,GTX, GTY); GTX+GTY;	00129300
	POSTAPE(LIBINIX);	00129400
	END;	00129500
AA2:	PFROMB;	00129600
	IF ADDINDEX < 3 THEN BEGIN	00129700
	IF NOT OPENED(ADDINDEX) THEN GO TO E5;	00129800
	IF LOOK(ACCUMA,NA,DIRECTORY, ADDINDEX * 3,	00129900
	GTX, GTY) THEN GO TO E4;	00130000
	POSTAPE(ADDINDEX); END;	00130100
A2:	IF RENAMETOG THEN BEGIN IF	00130200
	ENTERDIRECTORY(DIRECTORY,EDROW,EDROW,EDPOS,EDPOS,	00130300
	ACCUMB,NB,NEWRC) THEN ELSE GO TO E10;	00130400
	TESTMAKER+	00130500
	TAPEMAKER(ADDINDEX,ACCUMB,NB);	00130600
	IF TESTMAKER#0 THEN GO TO E8; GO TO ADDONE; END;	00130700
	IF REPLACETOG AND NOT WITHTOG THEN BEGIN	00130710
	IF ENTERDIRECTORY(DIRECTORY,EDROW,EDROW,EDPOS,EDPOS,	00130720
	ACCUMC, NC,NEWRC) THEN ELSE GO TO E10;	00130730
	IF TESTMAKER+TAPEMAKER(ADDINDEX,ACCUMC,NC) #0 THEN GO TO E8;	00130740
	GO TO ADDONE; END;	00130750
	IF ENTERDIRECTORY(DIRECTORY,EDROW,EDROW,EDPOS,EDPOS,	00130800
	ACCUMA,NA,NEWRC) THEN ELSE GO TO E10;	00130900
	TESTMAKER+	00131000
	TAPEMAKER(ADDINDEX,ACCUMA,NA);	00131100
	IF TESTMAKER#0 THEN GO TO E8; GO TO ADDONE;	00131200
ADDONE: GO TO MORE;		00131300
WRAPUP: IF MKETOG THEN		00131400
	BEGIN WHILE B1 DO	00131500
	BEGIN IF MPUNTOG THEN PPP,[46:1]+1;	00131600
	IF MLISTOG THEN PPP,[47:1]+1;	00131700
	B1+GETNEXTBLOB(DIRECTORY,ROW,ROW,II,II,	00131800
	ACCUMA,NA,GTX,GTY);	00131900
	IF NOT ENTERDIRECTORY(DIRECTORY,EDROW,EDROW,	00132000
	EDPOS,EDPOS,ACCUMA,NA,NEWRC) THEN BEGIN	00132100
	OUTPUTV+10; GO OUTPUT; END;	00132200
	POSTAPE(LIBINIX);	00132300
	TEMP+TAPEMAKER(LIBINIX,ACCUMA,NA);	00132400
	END;END;	00132500
	IF ENTERDIRECTORY(DIRECTORY,EDROW,EDROW,EDPOS,EDPOS,ACCUMA,	00132600
	O,NEWRC) THEN BEGIN END; COMMENT DUMMY ENTRY INTO	00132700
	ENTERDIRECTORY TO INSERT ENDING	00132800
	RECORD + 1 OF LAST BLOB;	00132900
	WRITE(FOUT[LIBOUTIX]);	00133000
	WRITE(FOUT[LIBOUTIX]);	00133100
	WRITE(FOUT[LIBOUTIX]);	00133200

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

Date Documents/In.

Data Documents, Inc.

1				00133300	1
2				00133400	2
3				00133500	3
4				00133600	4
5				00133700	5
6				00133800	6
7				00133900	7
8				00134000	8
9				00134100	9
10				00134200	10
11				00134300	11
12				00134400	12
13				00134500	13
14				00134600	14
15				00134700	15
16				00134800	16
17				00134900	17
18				00135000	18
19				00135100	19
20				00135200	20
21				00135300	21
22				00135400	22
23				00135500	23
24				00135600	24
25				00135700	25
26				00135800	26
27				00135900	27
28				00136000	28
29				00136100	29
30				00136200	30
31				00136300	31
32				00136400	32
33				00136500	33
34				00136600	34
35				00136700	35
36				00136800	36
37				00136900	37
38				00137000	38
39				00137100	39
40				00137200	40
41				00137300	41
42				00137400	42
43				99999999	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

REWIND (FOUT[LIBOUTIX]);

DIRECTORY[9,0]+3;

FOR TEMP+1 STEP 1 UNTIL DIRECTORY[9,0] DO
BEGIN MOVE(56,DIRECTORY[TEMP+8,0],FOUT[LIBOUTIX](0));
WRITE(FOUT[LIBOUTIX]);
END;

FIXFIB5(DUMY,LIBOUTIX);

IF DIRTOG THEN BEGIN
PAG; PRINTDIRECTORY(9); PAG; END;
IF NOT DISTOG AND COPYTOG THEN BEGIN

REWIND (FOUT[LIBOUTIX]);
WHILE TRUE DO BEGIN
READ (FOUT[LIBOUTIX],56, DIRECTORY[9,*])[EOF];
WRITE (CF [LIBOUTIX],56, DIRECTORY[9,*]);

END; END; EOF: GO TO E3;

END MAKEFUNCTION;

E0: OUTPUTV+0; GO OUTPUT;
E1: OUTPUTV+1; GO OUTPUT;
E2: OUTPUTV+2; GO OUTPUT;
E3: OUTPUTV+3; GO OUTPUT;
E4: OUTPUTV+4; GO OUTPUT;
E5: OUTPUTV+5; GO OUTPUT;
E6: OUTPUTV+6; GO OUTPUT;
E7: OUTPUTV+7; GO OUTPUT;
E8: OUTPUTV+8; GO OUTPUT;
E9: OUTPUTV+9; GO OUTPUT;
E10: OUTPUTV+10; GO OUTPUT;

E11: OUTPUTV+11; GO TO OUTPUT;

OUTPUT: PAG;

WRITE (LINE, SWF[OUTPUTV]);

LASTCARD: END.

END;END. LAST CARD ON OCRDING TAPE
00000000)x≥A40+PSTACK(F+13) = T3

0000000000000000)x≥A40+0

LABEL 000000000PRINTER00175100CC EX OBJECT/READ;FILE SOURCEFILE=SYMBOL/MAKCAST;END*

OBJECT /READ

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.