

LABEL 000000000PRINTER00175099CC EX OBJECT/READ;FILE SOURCEFILE=SYMBOL/PAPER;END*

OBJECT /READ

SYMBOL/PAPER

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

```

BEGIN REAL COMMON;
*..... CANDE PAPER TAPE PROGRAM - REVISED 3-72 (SHM).....
COMMENT: * TITLE: B5500/B5700 MARK XIV SYSTEM RELEASE *
* FILE ID: SYMBOL/PAPER 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 *;
FILE TWX 14 (2,11);
FILE SOURCE DISK SERIAL (2, 0, 0) ; % SOURCE FILE
FILE PATCH DISK SERIAL [20:600] (2,10,300,SAVE 1); % PATCH FILE
FILE WRKFIL DISK SERIAL [20:600] (2,10,300,SAVE 1); % WORK FILE
FILE NEWTAB DISK SERIAL [20: 30] (2,30,300,SAVE 1); % TAB FILE
SAVE ARRAY SORTA, MERGE[0:512], A,B,N[0:30], T[0:10];
ARRAY SORTAA, SORTBB[C:96,0:32];
LABEL COMPARE, ENDMERGE, EOF1, EOF4, EOF5, EXIT,
NEXTP, PCYCLE, READ1, STARTMERGE, SKIP,
TLOOP, WRITENEW;
BOOLEAN BFULL, BRAAK, EQLTOG, MERGTG,
SEQMODE, SFLG, SORTCG;
REAL A1, A2, A3, A4, AAPTR, AASEQ, AAWORD, ASEQ,
AWORD, B1, B2, B3, B4, B5, BASE, CODE,
EOFMARK, I, IA, IAA, IBB, INCR, J, LINE,
NCT, NPTR, NSEQ, NWORD, O1, O2, PATCNT,
N1, N2, PPTR, PREV, PSEQ, PWORD, RECSZ, SEQ, SNAME,
SZN, SZO, TSEQ, USER;
DEFINE LEFTARROW="←";
*****
STREAM PROCEDURE MOVE(N,A,B); VALUE N; % MOVE N WDS FROM A TO B
*****
BEGIN LOCAL M;
SI:=LOC N; DI:=LOC M; DI:=DI+1; DS:=7 CHR;
SI:=A; DI:=B; DS:=N WDS; M(DS:=32 WDS; DS:=32 WDS);
END;
*****
PROCEDURE SORT(L,U); % RECURSIVE SORT ROUTINE
*****
VALUE L,U;
REAL L,U;
BEGIN REAL I,J,K,M;
LABEL AGAIN, TOP, BOTTOM, EXIT;
IF L NEQ U THEN
IF L+1=U THEN
BEGIN IF SORTA[L],[21:27] GTR SORTA[U],[21:27] THEN
DOUBLE(SORTA[L],SORTA[U],←,SORTA[U],SORTA[L]);
END
ELSE
BEGIN M:=(U+L) DIV 2;
SORT(L,M); SORT(M+1,U);
J:=M+1;
FOR LI=L STEP 1 WHILE SORTA[LI],[21:27] LSS SORTA[J],[21:27] DO;
IF LI LEQ M THEN
BEGIN I:=KI=L;
AGAIN: IF I GTR M THEN GO TO TOP;
IF J GTR U THEN GO TO BOTTOM;
IF SORTA[I],[21:27] LEQ SORTA[J],[21:27] THEN GO TO BOTTOM;

```

```

TOP:    MERGE[K]:=SORTA[J];          00015000
        J:=J+1;                    00015100
        IF K:=K+1 LEG U THEN GO TO AGAIN ELSE GO EXIT; 00015200
BOTTOM: MERGE[K]:=SORTA[I];        00015300
        I:=I+1;                    00015400
        IF K:=K+1 LEG U THEN GO TO AGAIN; 00015500
EXIT:   MOVE(U-L+1,MERGE[L],SORTA[L]); 00015600
        END                          00015700
        END                          00015800
        END SORT;                   00015900
*****                                00016000
PROCEDURE DISKWAIT(L,A,S,D);        00016100
VALUE I,S,D; REAL I,S,D; ARRAY A[*]; COMMUNICATE(-8); 00016200
*****                                00016300
PROCEDURE IWXQUIT(A,N,T);          00016400
*****                                00016500
VALUE N,T;                          00016600
REAL A,N,T;                           00016700
BEGIN COMMUNICATE(-11);              00016800
BRAK := BOOLEAN(T); % MCP RETURNS 1 IF BREAK OCCURRED. 00016900
END;                                   00017000
*****                                00017100
PROCEDURE MSG(NO,SEQ); VALUE NO,SEQ; REAL NO,SEQ; 00017200
*****                                00017300
BEGIN REAL SIZ,SZ;                   00017400
REAL STREAM PROCEDURE NCONV(N,SIZ); VALUE N; 00017500
BEGIN                                00017600
SI:=LOC N; DI:=LOC NCONV; DS:=8DEC; DI:=DI-8; DS:=7FILL; 00017700
DI:=LOC NCONV; SI:=LOC NCONV; IF SC="" THEN 00017800
BEGIN                                00017900
8(IF SC="" THEN SI:=SI+1 ELSE 00018000
BEGIN DS:=CHR; TALLY:=TALLY+1; END); 00018100
END ELSE TALLY:=8;                   00018200
N:=TALLY; SI:=LOC N; DI:=SIZ; DS:=WDS; 00018300
END PROCEDURE NCONV;                 00018400
% .....                                00018500
REAL STREAM PROCEDURE M1(A,N,SIZ); VALUE N,SIZ; 00018600
BEGIN                                00018700
DI:=A; DS:=7LIT"RECORD "; SI:=LOC N; DS:=SIZ CHR; 00018800
DS:=9 LIT " DELETED."; TALLY:=16; M1:=TALLY; 00018900
END STREAM PROCEDURE M1;             00019000
% .....                                00019100
REAL STREAM PROCEDURE M2(A,N,SIZ); VALUE N,SIZ; 00019200
BEGIN                                00019300
DI:=A; DS:=7LIT"RECORD "; SI:=LOC N; DS:=SIZ CHR; 00019400
DS:=21LIT" IS NOT IN YOUR FILE."; TALLY:=28; M2:=TALLY; 00019500
END STREAM PROCEDURE M2;             00019600
% .....                                00019700
REAL STREAM PROCEDURE M3(A,N,SIZ); VALUE N,SIZ; 00019800
BEGIN                                00019900
DI:=A; DS:=24LIT"IMPROPER FIX FOR RECORD "; 00020000
SI:=LOC N; DS:=SIZ CHR; TALLY:=24; M3:=TALLY; 00020100
END STREAM PROCEDURE M3;             00020200
% .....                                00020300
REAL STREAM PROCEDURE M4(A,N,SIZ); VALUE N,SIZ; 00020400
BEGIN                                00020500
DI:=A; DS:=33LIT"MISSING GROUP MARK IN FIX RECORD "; 00020600
SI:=LOC N; DS:=SIZ CHR; TALLY:=33; M4:=TALLY; 00020700
END STREAM PROCEDURE M4;             00020800
% .....                                00020900

```

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

```

REAL STREAM PROCEDURE M5(A,N,SIZ); VALUE N,SIZ;          00021000
BEGIN                                                    00021100
  DI:=A; DS:=41LIT"CANNOT LOCATE YOUR FIX STRING FOR RECORD "; 00021200
  SI:=LOC N; DS:= SIZ CHR; TALLY:=41; M5:=TALLY;        00021300
END STREAM PROCEDURE M5;                                00021400
%.....00021500
REAL STREAM PROCEDURE M6(A,N,SIZ); VALUE N,SIZ;        00021600
BEGIN                                                    00021700
  DI:=A; DS:=39LIT"NOT ENOUGH ROOM FOR YOUR FIX IN RECORD "; 00021800
  SI:=LOC N; DS:= SIZ CHR; TALLY:=39; M6:=TALLY;        00021900
END STREAM PROCEDURE M6;                                00022000
%.....00022100
REAL STREAM PROCEDURE M7(A,N,SIZ); VALUE N,SIZ;        00022200
BEGIN                                                    00022300
  DI:=A; SI:=LOC N; DS:=SIZ CHR;                        00022400
  DS:=25LIT" EMPTY RECORDS DISCARDED."; TALLY:=25; M7:=TALLY; 00022500
END STREAM PROCEDURE M7;                                00022600
%.....00022700
REAL STREAM PROCEDURE M8(A,N,SIZ); VALUE N,SIZ;        00022800
BEGIN                                                    00022900
  DI:=A; SI:=LOC N; DS:= SIZ CHR;                       00023000
  DS:=46LIT" RECORDS DISCARDED (MISSING SEQUENCE NUMBERS)."; 00023100
  TALLY:=46; M8:=TALLY;                                  00023200
END STREAM PROCEDURE M8;                                00023300
%.....00023400
REAL STREAM PROCEDURE M9(A,N,SIZ); VALUE N,SIZ;        00023500
BEGIN                                                    00023600
  DI:=A; SI:=LOC N; DS:= SIZ CHR;                       00023700
  DS:=30LIT" RECORDS TRUNCATED (TOO LONG)."; TALLY:=30; M9:=TALLY; 00023800
END STREAM PROCEDURE M9;                                00023900
%.....00024000
REAL STREAM PROCEDURE M10(A);                          00024100
  BEGIN DI:=A; DS:=5LIT"WAIT."; TALLY:=5; M10:=TALLY; END; 00024200
%.....00024300
SEQ := NCONV(SEQ,SIZ);                                  00024400
IF NO=1 THEN SZ:=M1(A,SEQ,SIZ) ELSE                     00024500
IF NO=2 THEN SZ:=M2(A,SEQ,SIZ) ELSE                     00024600
IF NO=3 THEN SZ:=M3(A,SEQ,SIZ) ELSE                     00024700
IF NO=4 THEN SZ:=M4(A,SEQ,SIZ) ELSE                     00024800
IF NO=5 THEN SZ:=M5(A,SEQ,SIZ) ELSE                     00024900
IF NO=6 THEN SZ:=M6(A,SEQ,SIZ) ELSE                     00025000
IF NO=7 THEN SZ:=M7(A,SEQ,SIZ) ELSE                     00025100
IF NO=8 THEN SZ:=M8(A,SEQ,SIZ) ELSE                     00025200
IF NO=9 THEN SZ:=M9(A,SEQ,SIZ) ELSE                     00025300
  BEGIN SZ:=M10(A); SIZ:=0; END;                         00025400
  TWXOUT(A[0],(SIZ+SZ),1);                               00025500
END PROCEDURE MSG;                                       00025600
%*****00025700
DEFINE TRB=(IF SB THEN DS:=SET ELSE DS:=RESET;SKIP SB)#; 00025800
DEFINE FTC=SI:=SI+3; DI:=DI+5; SKIP 3 DB;15TRB#;        00025900
DEFINE CTF=SI:=SI+5; SKIP 3 SB; DI:=DI+3;15TRB#;        00026000
%*****00026100
STREAM PROCEDURE RETURN(A,B);                          00026200
%*****00026300
BEGIN LOCAL AT,BT;                                      00026400
SI:=LOC A; DI:=LOC AT; FTC;                             00026500
SI:=LOC B; DI:=LOC BT; FTC;                             00026600
SI:=LOC BT; DI:=LOC A; CTF;                             00026700
SI:=LOC AT; DI:=LOC B; CTF;                             00026800
DI:=AT; SI:=LOC B; DS:=WDS;                             00026900

```

```

DI:=BT; SI:=LOC A; DS:=WDS;                                00027000
END;                                                         00027100
%*****00027200
PROCEDURE SWAP;                                           00027300
%*****00027400
BEGIN                                                     00027500
  ARRAY TS, TM[0:1];                                       00027600
  RETURN(SORTA, TS); RETURN(MERGE, TM);                   00027700
END PROCEDURE SWAP;                                       00027800
%*****00027900
REAL STREAM PROCEDURE DECONV(X); VALUE X;                00028000
  BEGIN SI:=LOC X; DI:=LOC DECONV; DS:=8 DEC; END;       00028100
%*****00028200
REAL STREAM PROCEDURE OCTCONV(X);                       00028300
  BEGIN SI:=X; DI:=LOC OCTCONV; DS:=8 OCT; END;          00028400
%*****00028500
REAL STREAM PROCEDURE WHEREIS(X); % GET ADDRESS OF "X"   00028600
  BEGIN SI:=X; WHEREIS:=SI; END;                         00028700
%*****00028800
REAL PROCEDURE SIZ(A1, A2); VALUE A1, A2; REAL A1, A2; % GET SIZE FROM ADRS 00028900
SIZ:=(A2, [33:15]-A1, [33:15]) $\times$ 8+A2, [30:3]-A1, [30:3]; 00029000
%*****00029100
PROCEDURE TWXREAD(SFLG, SEQMODE); VALUE SFLG, SEQMODE; BOOLEAN SFLG, SEQMODE; 00029150
%*****00029200
BEGIN                                                     00029250
  % SFLG IS TRUE IF SEQUENTIAL TYPE FILE                 00029300
  % SEQMODE IS TRUE IF "TAPE SEQUENCE"                  00029350
  REAL LINKPTR, MINPTR, SEQPTR, MAXPTR, PTR1, PTR2, LPTR; 00029400
  REAL BLNKCNT, TRNCNT, ERRCNT, SEQNU, N, I, SIZE, SZ64, SZ1, TOKEN; 00029450
  LABEL READIN, ERR, FIXCK, EXIT;                      00029500
  BOOLEAN CKSEQ;                                       00029550
  %.....00029600
  REAL STREAM PROCEDURE FINDARROW(LINKPTR, MINPTR, MAXPTR, PTR2); 00029650
  %.....00029700
  VALUE LINKPTR, MINPTR, MAXPTR;                      00029750
  BEGIN LOCAL SV; LABEL L, EXIT;                      00029800
  % SEARCHES FOR GROUP MARK IN INPUT RECORD, RETURNS ADDRESS TO PTR2 00029810
  % LINKPTR IS ADDRESS OF FIRST CHARACTER BELOW START OF RCORD 00029850
  % MINPTR IS ADDRESS OF FIRST CHARACTER IN INPUT RECORD 00029900
  % MAXPTR IS ADDRESS OF MAXIMUM CHARACTER POSITION ALLOWED IN RECORD 00029950
  % RETURNS 0 FOR NO GROUP MARK ( TRUNCATED RECORD ) 00030000
  % RETURNS 1 FOR EMPTY RECORD ( GROUP MARK = FIRST CHARACTER ) 00030050
  % RETURNS 2 FOR VALID RECORD ( GROUP MARK NEQ FIRST CHARACTER ) 00030100
  % RETURNS 3 FOR END OF INPUT ( QMK = FIRST CHARACTER ) 00030150
  SI:=MAXPTR; % START AT END OF INPUT RECORD 00030200
  L: IF SC=LEFTARROW THEN % CHECK FOR POSITION OF GROUP MARK 00030250
  BEGIN 00030300
    SV:=SI; SI:=LOC SV; DI:=PTR2; DS:=WDS; % ADDRESS OF GROUP MARK 00030350
    SI:=LOC SV; SI:=SI+5; DI:=LOC LINKPTR; DI:=DI+5; 00030400
    IF 3SC=DC THEN GO TO EXIT; % TRUNCATED RECORD, RETURN ZERO 00030450
    SI:=SI-3; DI:=LOC MINPTR; DI:=DI+5; 00030500
    IF 3SC=DC THEN % EMPTY RECORD 00030550
      BEGIN TALLY:=1; GO TO EXIT; END; 00030600
    SI:=MINPTR; IF SC GTR "9" THEN TALLY:=3 ELSE TALLY:=2; 00030650
    GO TO EXIT; 00030700
  END; % IF GROUP MARK FOUND 00030750
  SI:=SI-1; GO TO L; % CONTINUE SEARCHING FOR GROUP MARK 00030800
EXIT: 00030850
  FINDARROW:=TALLY; % RETURN RESULT 00030900
  00030950

```

DI:=MAXPTR; DS:=LIT LEFTARROW; % PLACE MARKER AT END OF RECORD	00031000
END STREAM PROCEDURE FINDARROW;	00031050
%.....	00031100
1 REAL STREAM PROCEDURE NEXTOKEN(PTR1,PTR2,TOKEN); VALUE PTR2;	00031150
2 %.....	00031200
3 BEGIN LOCAL T,SV; LABEL L,EXIT;	00031250
4 % PTR1 IS ADDRESS OF NEXT CHARACTER IN INPUT RECORD	00031300
5 % PTR2 IS ADDRESS OF END OF INPUT RECORD	00031350
6 % RETURNS 0 FOR BLANK STRING	00031400
7	00031410
8	00031420
9	00031430
10 % RETURNS 1 FOR SPECIAL CHARACTER	00031450
11 % RETURNS 2 FOR NUMERIC STRING (SEQUENCE NUMBER)	00031500
12 % RETURNS 3 FOR ALPHABETIC STRING ("FIX" REQUEST)	00031550
13 % (PROGRAM ADJUSTS VALUE TO 4 IF A DELETE REQUEST)	00031600
14 SI:=PTR1; SI:=SI+5; SI:=SC; % POINT TO START OF INPUT STRING	00031650
15 L: IF SC=" " THEN	00031700
16 BEGIN SI:=SI+1; GO TO L; END;	00031750
17 IF SC=ALPHA THEN ELSE % SPECIAL CHARACTER	00031800
18 BEGIN	00031850
19 DI:=TOKEN; DS:=7LIT"0"; DS:=CHR; SI:=SI-1;	00031900
20 IF SC=LEFTARROW THEN % CHECK FOR END OF RECORD	00031950
21 BEGIN	00032000
22 SV:=SI; SI:=LOC SV; SI:=SI+5; DI:=LOC PTR2; DI:=DI+5;	00032050
23 IF 3SC NEQ DC THEN TALLY:=1; % NOT END OF RECORD	00032100
24 SI:=SV; GO TO EXIT;	00032150
25 END ELSE SI:=SI+1; % RESEQ INDEX IF NOT GROUP MARK	00032200
26 TALLY:=1; GO TO EXIT;	00032250
27 END; % IF SPECIAL CHARACTER	00032300
28 IF SC GEQ "0" THEN IF SC LEQ "9" THEN; % NUMERIC TOKEN	00032350
29 IF TOGGLE THEN	00032400
30 BEGIN	00032450
31 SV:=SI; SI:=SI+1; TALLY:=1;	00032500
32 7(IF SC GEQ "0" THEN IF SC LEQ "9" THEN;	00032550
33 IF TOGGLE THEN ELSE JUMP OUT; SI:=SI+1; TALLY:=TALLY+1);	00032600
34 T:=TALLY; SI:=SV; DI:=TOKEN; DS:=T OCT;	00032650
35 TALLY:=2; GO TO EXIT;	00032700
36 END; % IF NUMERIC TOKEN	00032750
37 % ALPHABETIC TOKEN	00032800
38 DI:=TOKEN; DS:=8LIT"0 " ; DI:=DI-7; DS:=CHR;	00032850
39 6(IF SC=ALPHA THEN IF SC LEQ "0" THEN;	00032900
40 IF TOGGLE THEN ELSE JUMP OUT; DS:=CHR);	00032950
41 TALLY:=3;	00033000
42 EXIT;	00033050
43 SV:=SI; SI:=LOC SV; DI:=PTR1; DS:=WDS; NEXTOKEN:=TALLY;	00033100
44 END STREAM PROCEDURE NEXTOKEN;	00033150
45 %.....	00033200
46 STREAM PROCEDURE PLACEMARKER(PTR); VALUE PTR;	00033250
47 %.....	00033300
48 BEGIN	00033350
49 DI:=PTR; DS:=LIT LEFTARROW;	00033400
50 END STREAM PROCEDURE PLACE MARKER;	00033450
51 %.....	00033500
52 STREAM PROCEDURE REFORMAT(PTR1,SZ64,SZ1,SFLG,SEQPTR,SEQNO,B);	00033550
53 %.....	00033600
54 VALUE PTR1,SZ64,SZ1,SFLG,SEQPTR,SEQNO;	00033650
55 BEGIN	00033700
56 % PTR1 IS ADDRESS OF FIRST TEXT CHARACTER IN INPUT RECORD	00033750
57 % SZ4 IS NO.OF CHRS TO BE TRANSFERRED DIV 64	00033800

```

% SZ1 IS NO. OF CHRS TO BE TRANSFERRED MOD 64          00033850
% SFLG=TRUE IF SEQ.NO. IS TO BE PLACED IN OUTPUT RECORD 00033900
% SEQPTR IS ADDRESS OF SEQ.NO. POSITION IN OUTPUT RECORD 00033950
% SEQNO IS OCTAL VALUE OF SEQUENCE NUMBER             00034000
% "B" IS OUTPUT ARRAY                                00034050
DI:=B; DS:=BLIT" "; SI:=B; DS:=9WDS;                 00034100
SI:=PTR1; DI:=B;                                     00034150
SZ64(2(DS:=32CHR)); DS:=SZ1 CHR;                     00034200
SFLG(SI:=LOC SEQNO; DI:=SEQPTR; DS:=8DEC);           00034250
END STREAM PROCEDURE REFORMAT;                        00034300
%.....00034350
DEFINE MATCH(MATCH1,MATCH2)=00034400
REAL(BOOLEAN(MATCH1) EQV BOOLEAN(MATCH2))=REAL(NOT FALSE)#; 00034450
%.....00034500
A[0]:=B[0]:=0; % MAKE SAVE ARRAYS PRESENT             00034550
MINPTR:=WHEREIS(A[0]); % ADDRESS OF START OF INPUT RECORD 00034600
LINKPTR:=(MINPTR-1)&7[30:45:3]; PLACEMARKER(LINKPTR); 00034650
SEQPTR:=WHEREIS(B[9]);                                00034700
CKSEQ:=SFLG AND NOT SEQMODE;                          00034750
MAXPTR:=MINPTR+9+                                     00034800
(IF NOT SFLG OR (SFLG AND NOT SEQMODE) THEN 1 ELSE 0); 00034850
PPTR := PATCNT := PREV := -1;                         00034900
READIN:                                                00034950
READ(TWX[STOP],11,A[*]); PTR1:=MINPTR;                00035000
IF (N:=FINDARROW(LINKPTR,MINPTR,MAXPTR,PTR2))=3 THEN GO TO EXIT; 00035050
IF N=1 THEN % EMPTY RECORD ( GROUP MARK ONLY )       00035100
BEGIN                                                 00035150
BLNKCNT:=BLNKCNT+1; GO TO READIN;                    00035200
END;                                                  00035250
IF N=0 THEN % TRUNCATED RECORD                       00035300
BEGIN                                                 00035350
TRNCNT:=TRNCNT+1; PTR2:=MAXPTR;                     00035400
END;                                                  00035450
IF CKSEQ THEN % SEQUENCE NUMBER SHOULD APPEAR IN INPUT RECORD 00035500
BEGIN                                                 00035550
IF (N:=NEXTOKEN(PTR1,PTR2,TOKEN))=0 THEN % BLANK RECORD 00035600
BEGIN                                                 00035650
ERR: ERRCNT:=ERRCNT+1; GO TO READIN;                 00035700
END;                                                  00035750
IF N=1 THEN % SPECIAL CHARACTER                     00035800
IF TOKEN NEQ "*" THEN GO TO ERR ELSE % ONLY "*" ACCEPTED 00035850
BEGIN N:=3; GO TO FIXCK; END; % MARK AS "FIX" REQUEST (N=3) 00035900
IF N=3 THEN % ALPHABETIC STRING                     00035950
IF MATCH(TOKEN,"FIX ") THEN                          00036000
BEGIN                                                00036050
FIXCK:IF (NEXTOKEN(PTR1,PTR2,TOKEN)) NEQ 2 THEN GO ERR; % NOT NUMERIC 00036100
PTR2:=(IF (I:=PTR2.[30:3])=7 THEN PTR2.[33:15]+1 ELSE 00036150
(PTR2&(I+1)[30:45:3])); % KEEP GROUP MARK IF A FIX
END ELSE GO TO ERR; % ALPHABETIC STRING NOT A "FIX" 00036200
SEQNO:=TOKEN;                                        00036250
IF N=2 THEN % NUMERIC STRING                         00036300
BEGIN                                                00036350
LPTR:=PTR1;                                          00036400
IF NEXTOKEN(LPTR,PTR2,TOKEN)=0 THEN % GROUP MARK 00036450
IF LPTR=PTR1 THEN N:=4; % DELETE REQUEST 00036500
END;                                                  00036550
END % IF CKSEQ                                       00036600
ELSE IF SEQMODE THEN SEQNO:=(BASE:=BASE+INCR);      00036650
IF N NEQ 4 THEN % NOT DELETE REQUEST                 00036700
BEGIN                                                00036750
00036800

```

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

```

SIZE:=(PTR2.[33:15]-PTR1.[33:15])*8+PTR2.[30:3]-PTR1.[30:3]; 00036850
SZ64:=SIZE.[36:6]; SZ1:=SIZE.[42:6]; 00036900
REFORMAT(PTR1,SZ64,SZ1,(SFLG AND (N NEQ 3)),SEQPTR,SEQNO,B); 00036950
WRITE(PATCH,10,B[*]); 00037000
END; 00037050
IF SFLG THEN % SEQUENTIAL TYPE FILE 00037100
BEGIN 00037150
PWORD:=SEQNO&N[1:45:3]&(PATCNT:=PATCNT+REAL(N NEQ 4))[4:32:16]; 00037200
PPTR:=PPTR + 1; 00037250
MERGTG := MERGTG OR (N NEQ 2); % DELETE (4) OR FIX (3) 00037300
SORTAA[PPTR,[35:8],PPTR,[43:5]]:=PWORD; 00037350
IF NOT SEQMODE THEN IF SEQNO LEQ PREV THEN SORTG := TRUE; 00037400
PREV := SEQNO; 00037450
END ELSE WRITE(WRKFIL,10,B[*]); % DATA FILE 00037500
GO TO READIN; 00037600
EXIT; 00037650
TWXOUT(A[0],0,1); MSG(10,0); % "WAIT," 00037700
IF BLNKCNT GTR 0 THEN MSG(7,BLNKCNT); % "EMPTY RECORDS DISCARDED" 00037750
IF ERRCNT GTR 0 THEN MSG(8,ERRCNT); % "MISSING SEQ. NUMBERS" 00037800
IF TRNCNT GTR 0 THEN MSG(9,TRNCNT); % "TRUNCATED RECORDS" 00037850
END PROCEDURE TWXREAD; 00037900
%*****00038000
BOOLEAN STREAM PROCEDURE FIXCHK(A1,01,02,N1,N2,A4); VALUE A1,A4; 00038100
%*****00038200
COMMENT A1 IS ADDRESS OF A[0], 00038300
01,02 ARE START/FINISH ADDRESS OF "OLD STRING", 00038400
N1,N2 ARE START/FINISH ADDRESS OF "NEW STRING". 00038500
A4 IS ADDRESS OF END OF INPUT RECORD 00038600
ROUTINE EXAMINES "FIX" SYNTAX AND ESTABLISHES VALUES FOR 00038700
01,02,N1 AND N2; 00038800
BEGIN LOCAL SV,D,A3; LABEL L1,EXIT; 00038900
SI:=A4; 2(40(SI:=SI-1; IF SC=LEFTARROW THEN JUMP OUT 2 TO L1)); 00039000
L1: A3:=SI; SI:=A1; DI:=LOC D; 00039010
20(IF SC EQL " " THEN SI:=SI+1 ELSE JUMP OUT); % SCAN TO CHAR, 00039100
IF TOGGLE THEN GO TO EXIT; % EMPTY FIELD 00039200
DI:=LOC D; DI:=DI+7; DS:=CHR; % SAVE DELIMITER 00039300
IF SC=D THEN GO TO EXIT; % NO STRING 00039400
IF SC = LEFTARROW THEN 00039500
BEGIN 00039600
SV:=SI; SI:=LOC SV; SI:=SI+5; DI:=LOC A3; DI:=DI+5; 00039700
IF 3SC=DC THEN % LEFT ARROW REPLACEMENT 00039800
BEGIN 00039900
SI:=SI-1; SV:=SI; SI:=LOC SV; DI:=01; DS:=WDS; SI:=SV; 00040000
SI:=SI+1; SV:=SI; SI:=LOC SV; DI:=02; DS:=WDS; 00040100
SI:=LOC SV; DI:=N1; DS:=WDS; SI:=SV; 00040200
SI:=SI+1; SV:=SI; SI:=LOC SV; DI:=N2; DS:=WDS; 00040300
TALLY:=1; FIXCHK:=TALLY; GO TO EXIT; 00040400
END LEFT ARROW REPLACEMENT; 00040500
SI:=SV; 00040600
END; % IF LEFTARROW 00040700
SV:=SI; SI:=LOC SV; DI:=01; DS:=WDS; SI:=SV; % START OF OLD STRING 00040800
63(IF SC=LEFTARROW THEN % CHECK FOR END OF RECORD 00040900
BEGIN 00041000
SV:=SI; SI:=LOC SV; SI:=SI+5; DI:=LOC A3; DI:=DI+5; 00041100
IF 3SC=DC THEN JUMP OUT TO EXIT; 00041200
SI:=SV; 00041300
END; 00041400
IF SC NEQ D THEN SI:=SI+1 ELSE JUMP OUT); 00041500
IF TOGGLE THEN GO TO EXIT; % MISSING 2ND DELIMITER 00041600
SV:=SI; SI:=LOC SV; DI:=02; DS:=WDS; SI:=SV; %END OLD STRING 00041700

```

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.

```
SI:=SI+1; SV:=SI; SI:=LOC SV; DI:=N1; DS:=WDS; SI:=SV; % NEW STRING 00041800
63(IF SC=LEFTARROW THEN % CHECK FOR END OF RECORD 00041900
  BEGIN 00042000
  SV:=SI; SI:=LOC SV; SI:=SI+5; DI:=LOC A3; DI:=DI+5; 00042100
  IF 3SC=DC THEN JUMP OUT ELSE SI:=SV; 00042200
  END; 00042300
  SI:=SI+1); 00042400
SI:=LOC SV; DI:=N2; DS:=WDS; % END NEW STRING 00042500
TALLY:=1; FIXCHK:=TALLY; 00042600
EXIT: 00042700
  END STREAM PROCEDURE FIXCHK; 00042800
%*****00042900
BOULLEAN STREAM PROCEDURE LOCATE(B1,B2,B3,B4,L,O1,SZ0); 00043000
%*****00043100
COMMENT B1 IS ADDRESS OF B[0], 00043200
B2 IS ADDRESS OF START OF "EXISTING" STRING, 00043300
B3 IS ADDRESS OF CHARACTER FOLLOWING LAST NON-BLANK CHR. IN RECORD, 00043400
B4 IS ADDRESS OF B[9] IF NOT TYPE DATA, OF B[10] IF TYPE DATA 00043500
L IS RECORD SIZE, (2x)36 CHR. IF NOT TYPE DATA, (2x)40 CHR. IF TYPE DATA. 00043600
ROUTINE MATCHES "OLD STRING" AT ADDRESS O1, LENGTH SZ0 WITH 00043700
CONTENTS OF RECORD FOR "FIX" STATEMENT, AND SETS B2 AND B3 IF MATCHED; 00043800
VALUE B1,B4,L,O1,SZ0; 00043900
  BEGIN LOCAL SV,D; LABEL L1,L2,EXIT; 00044000
  SI:=O1; DI:=LOC O; DI:=DI+7; DS:=CHR; % SAVE FIRST CHARACTER 00044100
  SI:=B1; 00044200
  2(L(IF SC=D THEN 00044300
    BEGIN 00044400
      SV:=SI; DI:=O1; IF SZ0 SC = DC THEN 00044500
      BEGIN 00044600
        SI:=LOC SV; DI:=B2; DS:=WDS; JUMP OUT 2 TO L1; 00044700
      END; 00044800
      SI:=SV; 00044900
    END; 00045000
  SI:=SI+1)); 00045100
  GO TO EXIT; 00045200
L1: SI:=B4; SI:=SI-1; 00045300
  2(L(IF SC=" " THEN SI:=SI-1 ELSE JUMP OUT 2 TO L2)); 00045400
L2: SI:=SI+1; SV:=SI; SI:=LOC SV; DI:=B3; DS:=WDS; 00045500
  TALLY:=1; LOCATE:=TALLY; 00045600
EXIT: 00045700
  END STREAM PROCEDURE LOCATE; 00045800
%*****00045900
STREAM PROCEDURE EDIT(B2,B4,B5,SZ0,N1,SZN,SFLG); 00046000
%*****00046100
COMMENT B2 IS STARTING ADDRES OF "EXISTING" STRING, 00046200
B4 IS ADDRESS OF B[9] IF NOT TYPE DATA, OF B[10] IF TYPE DATA, 00046300
B5 IS ADDRESS OF B[20], 00046400
SZ0 IS LENGTH OF "EXISTING" STRING, 00046500
N1 IS STARTING ADDRESS OF REPLACEMENT STRING, 00046600
SZN IS LENGTH OF REPLACEMENT STRING, 00046700
SFLG IS TRUE IF NOT TYPE DATA FILE. 00046800
ROUTINE REPLACES "OLD" STRING WITH "NEW" STRING FOR "FIX"; 00046900
VALUE B2,B4,B5,SZ0,N1,SZN,SFLG; 00047000
  BEGIN LOCAL SEQ; 00047100
  SFLG(SI:=B4; DI:=LOC SEQ; DS:=WDS); % SAVE SEQUENCE NO, 00047200
  DI:=B4; 2(DS:=40LIT" "); 00047300
  SI:=B2; SI:=SI+SZ0; DI:=B5; 2(DS:=40CHR); 00047400
  SI:=N1; DI:=B2; DS:=SZN CHR; 00047500
  SI:=B5; 2(DS:=40CHR); 00047600
  SFLG(SI:=LOC SEQ; DI:=B4; DS:=WDS); 00047700
```

```

END STREAM PROCEDURE EDIT;                                00047800
*****00047900
STREAM PROCEDURE OUTFORMAT(A1,B1,B4,SQSH,SFLG);           00048000
*****00048100
COMMENT A1 IS ADDRESS OF A[0], B1 IS ADDRESS OF B[0],
B4 IS ADDRESS OF B[9] IF NOT TYPE DATA, OF B[10] IF TYPE DATA,
SQSH IS TRUE IF "SQUASHED" OPTION SET,
SFLG IS TRUE IF NOT TYPE DATA FILE,
ROUTINE REFORMATS RECORD FOR OUTPUT TO REMOTE STATION;
VALUE A1,B1,B4,SQSH,SFLG;                                00048700
  BEGIN LOCAL SV,SEQ; LABEL EXIT;                        00048800
    DI:=A1; DS:=8LIT" "; SI:=A1; DS:=9WDS; DI:=A1;      00048900
    SFLG(SI:=B4; DI:=LCC SEQ; DS:=WDS; % SAVE SEC. NUMBER 00049000
    DI:=LOC SEQ; DS:= 7 FILL; SI:=LOC SEQ; DI:=A1;      00049100
    B(IF SC=" " THEN SI:=SI+1 ELSE DS:=CHR));            00049200
    DS:=LIT" "; SV:=DI; DI:=B4; DS:=8LIT" "; DI:=SV);    00049300
    SI:=B1;                                              00049400
    SQSH(                                                 00049500
      2(40( IF SC=" " THEN                                00049600
        BEGIN SI:=SI+1; IF SC NEQ " " THEN DS:=LIT" "; END 00049700
      ELSE DS:=CHR));                                     00049800
    JUMP OUT TO EXIT);                                    00049900
    2(DS:=40CHR); DS:=LIT" ";                             00050000
EXIT:                                                     00050100
  END STREAM PROCEDURE OUTFORMAT;                         00050200
*****00050300
*****00050400
EOFMARK:=100000000;                                     00050500
A[0]:=0; A1:=WHEREIS(A[0]); % SET "A" ADDRESSES           00050600
DISKWAIT(1,A,30,COMMON); % GET ESP RECORD                00050700
LINE:=DECONV(A[1],[40:8]); % DECIMAL VALUE OF LINE NO    00050800
SNAME := " " & "1S"[6:36:12] & LINE[18:30:18];          00050900
SFLG := A[1],[3:11]=0; % ON IF TYPE DATA FILE           00051000
B[0]:=0; B1:=WHEREIS(B[0]); % "B" ADDRESSES              00051100
B4:=WHEREIS(B[10]=REAL(SFLG)); B5:=WHEREIS(B[20]);       00051200
RECSZ:= 40 - 4 x REAL(SFLG); % 1/2 RECORD LENGTH        00051300
USER := A[2];                                             00051400
IF SEQMODE := A[7] GTR 0 THEN % SEQUENCE MODE OPERATION 00051500
  BEGIN                                                    00051600
    INCR := A[8],[21:27]; % SEQUENCE INCREMENT           00051700
    BASE := A[7],[21:27] - INCR;                          00051800
  END;                                                     00051900
A4 := A1 + 9 +                                           00052000
  (IF NOT SFLG OR (SFLG AND NOT SEQMODE) THEN 1 ELSE 0); 00052020
FILL NEWTAB WITH SNAME & "T"[12:42:6],USER; % "1T" TAB FILE 00052100
FILL PATCH WITH SNAME,USER; % "1S" PATCH FILE           00052200
FILL WRKFIL WITH SNAME,USER; % "1S" WORK FILE           00052300
FILL SOURCE WITH A[3],USER; % SOURCEFILE NAME           00052400
IF NOT SFLG THEN % TYPE DATA FILE                      00052500
  BEGIN                                                    00052600
    DO BEGIN % TRANSFER RECORDS FROM SOURCE, IF ANY      00052700
      READ(SOURCE,10,A[*])[EOF1];                        00052800
      WRITE(WRKFIL,10,A[*]);                              00052900
    END UNTIL FALSE;                                     00053000
  END;                                                    00053100
EOF1:                                                     00053200
CLOSE(SOURCE); TWXREAD(FALSE,FALSE);                    00053300
LOCK(WRKFIL,*);                                         00053400

```

00054700

```

GO TO EXIT;                                00054800
END; % IF TYPE DATA FILE                   00054900
TWXREAD(SFLG,SEQMODE);                      00058400
IF NOT SORTOG THEN GO TO STARTMERGE; % NO SORT REQUIRED 00058500
% .....START SORT ROUTINE.....            00058600
AAPTR := IF PPTR GTR 511 THEN 511 ELSE PPTR; 00058700
FOR I := 0 STEP 32 UNTIL AAPTR DO           00058800
FOR J := 0 STEP 1 UNTIL 31 DO              00058900
SORTA[I+J]:=SORTA[ I,[35:8],J ]; % MOVE TO SORTA, 512 WDS. MAX 00059000
IF AAPTR GTR 0 THEN SORT(0,AAPTR); % CALL SORT ROUTINE 00059100
FOR I := 0 STEP 1 UNTIL AAPTR DO           00059200
SORTA[ I,[35:8] , I,[43:5] ] := SORTA[I]; % MOVE BACK TO SORTAA 00059300
WHILE AAPTR LSS PPTR DO % SORT/MERGE REMAINDER IF PPTR GTR 511 00059400
BEGIN                                       00059500
N2 := (N1:=AAPTR+1) + 511;                00059600
IF N2 GTR PPTR THEN N2 := PPTR;           00059700
FOR I := N1 STEP 32 UNTIL N2 DO           00059800
FOR J := 0 STEP 1 UNTIL 31 DO            00059900
SORTA[I+J-N1] := SORTA[ I,[35:8] , J ]; % MOVE TO SORTA 00060000
IF (N2:=N2-N1) GTR 0 THEN SORT(0,N2);    00060100
IA := IAA := 0; IBB := -1; % MERGE SORTA WITH SORTAA 00060200
ASEQ := (AWORD := SORTA[IA]) .[21:27];   00060300
AASEQ := (AAWORD := SORTAA[ IAA,[35:8] , IAA,[43:5] ] ) .[21:27]; 00060400
DO BEGIN                                   00060500
IF AASEQ LEQ ASEQ THEN % USER OLDEST ONE FIRST 00060600
BEGIN                                       00060700
SORTBB[ (IBB:=IBB+1),[35:8] , IBB,[43:5] ] := AAWORD; 00060800
IF IAA:=IAA+1 GTR AAPTR THEN AASEQ:=EOFMARK ELSE 00060900
AASEQ:=(AAWORD:=SORTAA[ IAA,[35:8] , IAA,[43:5] ] ) .[21:27]; 00061000
END                                         00061100
ELSE                                       00061200
BEGIN                                       00061300
SORTBB[ (IBB:=IBB+1),[35:8] , IBB,[43:5] ] := AWORD; 00061400
IF IA:=IA+1 GTR N2 THEN ASEQ:=EOFMARK ELSE 00061500
ASEQ := (AWORD := SORTA[IA]) .[21:17];   00061600
END;                                       00061700
END UNTIL ASEQ = EOFMARK AND AASEQ = EOFMARK, 00061800
AAPTR := AAPTR + N2 + 1; % TOTAL NO. IN SORTAA 00061900
FOR I:=0 STEP 32 UNTIL AAPTR DO % MOVE BACK TO SORTAA 00062000
READ(SORTBB[ I,[35:8] , * ] , 32 , SORTAA[ I,[35:8] , * ] ); 00062100
END WHILE STATEMENT;                      00062200
SWAP; % RETURN CORE SPACE                  00062300
% .....                                00062400
STARTMERGE:                               00062500
% .....                                00062600
SORTAA[ (PPTR:=PPTR+1),[35:8], PPTR,[43:5] ] := EOFMARK; 00062700
READ(SOURCE[NO],10,T[*])[EOF4]; % TEST EOF POINTER 00062800
GO TO SKIP;                                00062900
% ....                                     00063000
EOF4; % NO RECORDS IN SOURCE FILE ( EOF POINTER = -1 ); 00063100
% ....                                     00063200
IF NOT (SORTOG OR MERGTOG) THEN % NO ALTERATIONS REQUIRED 00063300
BEGIN                                       00063400
CLOSE(SOURCE); LOCK(PATCH,*); % PATCH := WORKFILE 00063500
NOJ:=PPTR-1; % WORKFILE EOF POINTER 00063600
FOR I:=0 STEP 1 UNTIL PPTR DO % BUILD TAB FILE 00063700
BEGIN                                       00063800
IF NPTR:=NPTR+1 GTR 29 THEN % WRITE TABFILE SEGMENT ON DISK 00063900
BEGIN                                       00064000
WRITE(NEWTAB,30,N[*]);                    00064100

```

```

        NPTR:=0;                                00064200
        END;                                    00064300
        N(NPTR):=SORTAA(I.[35:8],I.[43:5]),[21:27] & I[4:32:16]; 00064400
    1  END ILOOP;                                00064500
    2  WRITE(NEWTAB,30,N[*]); LUCK(NEWTAB,*);    00064600
    3  GO TO EXIT;                               00064700
    4  END; % IF PATCH FILE BECOMES WORKFILE    00064800
    5  %....                                     00064900
    6  SKIP;                                     00065000
    7  %....                                     00065100
    8  PSEQ:=(PWORD:=SORTAA(0,0)).[21:27]; % 1ST SEQ. NUMBER IN "1P" FILE 00065200
    9  CODE:=PWORD.[1:3]; PPTR:=0;              00065300
   10 NCT := -1;                                00065400
   11 REWIND(PATCH); PATCNT := -1;              00065500
   12 %.....                                     00065600
   13 TLOOP:                                    00065700
   14 %.....                                     00065800
   15 READ(SOURCE,10,T[*])(EOF5);              00065900
   16 TSEQ := OCTCONV(T[9]);                    00066000
   17 GO TO COMPARE;                            00066100
   18 %....                                     00066200
   19 EOF5: TSEQ := EOFMARK;                    00066300
   20 %....                                     00066400
   21 %.....                                     00066500
   22 COMPARE:                                  00066600
   23 %.....                                     00066700
   24 IF TSEQ LSS PSEQ THEN % NO CHANGE TO THIS RECORD 00066800
   25 BEGIN                                     00066900
   26 WRITE(WRKFIL,10,T[*]);                    00067000
   27 EQLTOG:=TRUE; SEQ:=TSEQ; GO TO WRITENEW; 00067100
   28 END;                                       00067200
   29 EQLTOG := TSEQ=PSEQ; % NEW RECORD SEQ. MATCHES OLD 00067300
   30 %.....                                     00067400
   31 PCYCLE:                                   00067500
   32 %.....                                     00067600
   33 IF PSEQ=EOFMARK THEN GO TO ENDMERGE;      00067700
   34 IF CODE=4 THEN % DELETE CODE              00067800
   35 BEGIN                                     00067900
   36 BFULL:=FALSE;                             00068000
   37 PREV:=PSEQ; % SAVE CURRENT PSEQ NUMBER    00068100
   38 PPTR:=PPTR+1; % GET THE NEXT PSEQ NUMBER  00068200
   39 PSEQ:=(PWORD:=SORTAA(PPTR,[38:5],PPTR.[43:5]),[21:27]); 00068300
   40 CODE:=PWORD.[1:3];                        00068400
   41 IF PSEQ=PREV AND CODE=2 THEN GO PCYCLE; % NEW RECORD FOLLOWS 00068500
   42 IF EQLTOG THEN GO TO TLOOP ELSE GO TO COMPARE; 00068600
   43 END; % IF CODE = 4                          00068700
   44 IF CODE=3 THEN % FIX CODE, CHECK FOR RECORD FIRST 00068800
   45 BEGIN                                     00068900
   46 IF NOT (EQLTOG OR BFULL) THEN % NO SUCH RECORD 00069000
   47 BEGIN                                     00069100
   48 MSG(2,PSEQ); % "NOT IN YOUR FILE"         00069200
   49 GO TO NEXTP;                              00069300
   50 END;                                       00069400
   51 END;                                       00069500
   52 IF PATCNT := PATCNT + 1 NEQ PWORD.[4:16] THEN % OUT OF ORDER 00069600
   53 READ SEEK (PATCH[PATCNT:=PWORD.[4:16]]); 00069700
   54 IF CODE NEQ 3 THEN % NOT A FIX COMMAND     00069800
   55 BEGIN                                     00069900
   56 READ(PATCH,10,B[*]);                       00070000
   57 BFULL := TRUE;                             00070100

```

```

GO TO NEXTP;                                00070200
END;                                          00070300
READ(PATCH,10,A[*]);                          00070400
1 IF NOT FIXCHK(A1,01,02,N1,N2,A4) THEN % "FIX" SYNTAX ERROR 00070500
2 BEGIN                                       00070600
3 MSG(3,PSEQ); % "IMPROPER FIX"              00070700
4 GO TO NEXTP;                                00070800
5 END;                                          00070900
6 IF N2,[33:15] GTR A4 THEN % SHOULD NOT BE HERE 00071000
7 BEGIN                                       00071100
8 MSG(4,PSEQ); % "MISSING GROUP MARK"        00071200
9 GO TO NEXTP;                                00071300
10 END;                                         00071400
11 IF NOT BFULL THEN % GET RECORD TO BE "FIXED" 00071500
12 BEGIN                                       00071600
13 MOVE(10,T,B);                              00071700
14 BFULL:=TRUE;                               00071800
15 END;                                         00071900
16 IF NOT LOCATE(B1,B2,B3,B4,RECSZ,01,SZ0:=SIZ(01,02)) THEN 00072000
17 BEGIN                                       00072100
18 MSG(5,PSEQ); % "CANNOT FIND FIX STRING"    00072200
19 GO TO NEXTP;                                00072300
20 END;                                         00072400
21 IF SIZ(B3,B4) LSS (SZN:=SIZ(N1,N2))-SZ0 THEN 00072500
22 BEGIN                                       00072600
23 MSG(6,PSEQ); % "NOT ENOUGH ROOM FOR FIX"   00072700
24 GO TO NEXTP;                                00072800
25 END;                                         00072900
26 EDIT(B2,B4,B5,SZ0,N1,SZN,SFLG); % REPLACE OLD STRING WITH NEW STRING 00073000
27 %.....                                     00073100
28 NEXTP: % LOOK AT NEXT "1P" WORD             00073200
29 %.....                                     00073300
30 PREV:=PSEQ; % SAVE CURRENT "1P" SEQUENCE NUMBER 00073400
31 PPTR:=PPTR+1; % CHECK NEXT "1P" WORD        00073500
32 PSEQ:=(PWORD:=SORTAA[PPTR,[38:5],PPTR,[43:5]]).[21:27]; % NEXT "1P" WD. 00073600
33 CODE:=PWORD.[1:3];                          00073700
34 IF PREV=PSEQ THEN GO TO PCYCLE; % DUPLICATE, MORE TO BE DONE 00073800
35 IF NOT BFULL THEN GO TO COMPARE; % IGNORE LAST INPUT 00073900
36 WRITE(WRKFIL,10,B[*]);                       00074000
37 SEQ:=PREV; % USING THIS SEQ. NUMBER         00074100
38 %.....                                     00074200
39 WRITENEW;                                    00074300
40 %.....                                     00074400
41 IF NPTR:=NPTR+1 GTR 29 THEN % SEGMENT IS FILLED 00074500
42 BEGIN                                       00074600
43 WRITE(NEWTAB,30,N[*]); % OUTPUT TO DISK     00074700
44 NPTR:=0;                                    00074800
45 END;                                         00074900
46 N[NPTR]:=0&SEQ[21:21:27]&(NCT:=NCT+1)[4:32:16]; 00075000
47 BFULL:=FALSE;                               00075100
48 IF EQLT0G THEN GO TL0CP ELSE GO COMPARE;    00075200
49 %.....                                     00075300
50 ENDMERGE;                                    00075400
51 %.....                                     00075500
52 IF NPTR:=NPTR+1 GTR 29 THEN % SEGMENT IS FILLED 00075600
53 BEGIN                                       00075700
54 WRITE(NEWTAB,30,N[*]); % OUTPUT TO DISK     00075800
55 NPTR:=0;                                    00075900
56 END;                                         00076000
57 FOR I:=NPTR STEP 1 UNTIL 29 DO N[I]:=EOFMARK; 00076100

```

WRITE(NEWTAB,30,N[*]);
CLOSE(SOURCE); CLOSE(PATCH);
LOCK(NEWTAB,*); LOCK(WRKFIL,*);

%....
EXIT:
%....
END PROGRAM.
END;END.

LAST CARD ON OCRDING TAPE

165X219Y133D01*166X213Y133D02*167X213Y131D01*168X183Y131D01* 23071053s1

00076200
00076600
00076700
00076800
00076900
00077000
00077100
99999999

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 Documentis, Inc.

LABEL 00000000PRINTER00175099CC EX OBJECT/READ;FILE SOURCEFILE=SYMBOL/PAPER;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.