

LABEL 000000000PRINTER00175100CC EX OBJECT/READ;FILE SOURCEFILE=SYMBOL/HELP;END+

OBJECT /READ

SYMBOL/HELP

Data Documents/Inc.

```

BEGIN REAL COMMON;
COMMENT: * TITLE: B5500/B5700 MARK XV.1 SYSTEM RELEASE * 00100000
        * FILE ID: SYMBOL/HELP TAPE ID: SYMBOL2/FILE000 * 00100010
        * THIS MATERIAL IS PROPRIETARY TO BURROUGHS CORPORATION * 00100011
        * AND IS NOT TO BE REPRODUCED, USED, OR DISCLOSED * 00100012
        * EXCEPT IN ACCORDANCE WITH PROGRAM LICENSE OR UPON * 00100013
        * WRITTEN AUTHORIZATION OF THE PATENT DIVISION OF * 00100014
        * BURROUGHS CORPORATION, DETROIT, MICHIGAN 48232 * 00100015
        * * 00100016
        * * 00100017
        * COPYRIGHT (C) 1973, BURROUGHS CORPORATION * ; 00100018
FILE TWX 14(2,10); 00100100
SAVE ARRAY A,B[0:30], 00100500
BASE[0:16], T[0:30]; 00100600
ARRAY P[0:25]; %STORE INPUT PARAMETERS 00100700
ARRAY WORD[0:1], SDATA[0:30]; 00100800
SAVE ARRAY SORTA,MERGE[0:512]; 00100900
REAL ONAME, SRCENAME, VARNAME; BOOLEAN WRKFILE; 00101000
REAL A1, A2, A4, B1, B2, B3, B4, B5, EADRS, 00101100
EOFMARK,I,LINE,N1,N2,C1,C2,RECSZ,SEQ,USER; 00101200
BOOLEAN SFLG; 00101300
LABEL QUTR,QUTRMOST; 00101400
MONITOR FLAG; %CHECK FOR FLAG BIT ERRORS 00101500
INTEGER MAXCNT; 00101600
DEFINE LEFTARROW = "←"; 00101700
%*****
PROCEDURE GETHEADER(AREA,N1,N2,N3); 00101710
%***** 00101720
VALUE N1,N2,N3; REAL N1,N2,N3; ARRAY AREAL*; COMMUNICATE(-21); 00101730
%***** 00101740
STREAM PROCEDURE MOVETYPE(TYPE,FIIL); 00101800
%***** 00101900
VALUE TYPE; 00102000
BEGIN 00102100
SI:=FIIL; SI:=SI-24; DI:=LOC FIIL; DS:=WDS; 00102200
SI:=FIIL; 2(SI:=SI+56); DI:=LOC FIIL; DS:=WDS; 00102300
DI:=FIIL; DI:=DI+38; 00102400
SI:=LOC TYPE; SI:=SI+7; DS:=CHR; 00102500
END; 00102600
%***** 00102700
STREAM PROCEDURE MOVE(N,A,B); VALUE N; 00102800
%***** 00102900
BEGIN LOCAL M; 00103000
SI:=LOC N; DI:=LOC M; DI:=DI+1; DS:=7 CHR; 00103100
SI:=A; DI:=B; DS:=N WDS; M(DS:=32 WDS; DS:=32 WDS); 00103200
END; 00103300
%***** 00103400
STREAM PROCEDURE RESETSIZE(DSK,RSIZE,BSIZE,MAXROWS,ROWSIZE); 00103500
%***** 00103600
VALUE RSIZE,BSIZE,MAXROWS,ROWSIZE; 00103700
BEGIN 00103800
SI:=DSK; 3(SI:=SI-8); DI:=LOC DSK; DS:=WDS; 00103900
DI:=DSK; 8(DI:=DI+8); DI:=DI+3; SKIP 2DB; 00104000
SI:=LOC MAXROWS; SI:=SI+7; SKIP SB; 00104100
5(IF SB THEN DS:=SET ELSE DS:=RESET; SKIP SB); 00104200
SI:=LOC ROWSIZE; SI:=SI+4; SKIP SB; 00104300
23(IF SB THEN DS:=SET ELSE DS:=RESET; SKIP SB); 00104400
DI:=DSK; 18(DI:=DI+8); SKIP 3DB; 00104500
SI:=LOC BSIZE; SI:=SI+5; SKIP 3SB; 00104600
15(IF SB THEN DS:=SET ELSE DS:=RESET; SKIP SB); 00104700
SI:=LOC BSIZE; SI:=SI+5; SKIP 3SB; 00104800
SI:=LOC BSIZE; SI:=SI+5; SKIP 3SB; 00104900

```

```

15(IF SB THEN DS:=SET ELSE DS:=RESET; SKIP SB);          00105000
SI:=LOC RSIZE; SI:=SI+5; SKIP 3SB;                       00105100
15(IF SB THEN DS:=SET ELSE DS:=RESET; SKIP SB);          00105200
END RESETSIZE;                                           00105300
*****00105400
REAL STREAM PROCEDURE SIZEOFINPUT(A2); VALUE A2;         00105500
*****00105600
BEGIN                                                    00105700
% ROUTINE OBTAINS NUMBER OF CHARACTERS IN USERS INPUT RECORD 00105800
% FROM THE WORD WHICH PRECEEDS THE ACTUAL USERS INPUT.      00105900
% THE ACTUAL USERS INPUT BEGINS AT ADDRESS "A2"             00106000
SI:=A2; SI:=SI-3; DI:=LOC SIZEOFINPUT; DI:=DI+5; DS:=3CHR; 00106100
DI:=DI-3; DS:=3RESET;                                     00106200
END STREAM PROCEDURE SIZEOFINPUT;                         00106300
*****00106400
PROCEDURE SORT(L,U); % RECURSIVE SORT ROUTINE            00106500
*****00106600
VALUE L,U;                                               00106700
REAL L,U;                                                 00106800
BEGIN REAL I,J,K,M;                                       00106900
LABEL AGAIN, TOP, BOTTOM, EXIT;                          00107000
IF L NEQ U THEN                                          00107100
IF L+1=U THEN                                           00107200
BEGIN IF SORTA[L].[21:27] GTR SORTA[U].[21:27] THEN     00107300
DOUBLE(SORTA[L],SORTA[U],+,SORTA[U],SORTA[L]);         00107400
END                                                       00107500
ELSE                                                      00107600
BEGIN M:=(U+L) DIV 2;                                    00107700
SORT(L,M); SORT(M+1,U);                                  00107800
J:=M+1;                                                  00107900
FOR L:=L STEP 1 WHILE SORTA[L].[21:27] LSS SORTA[J].[21:27] DO; 00108000
IF L LEQ M THEN                                         00108100
BEGIN I:=K:=L;                                         00108200
AGAIN: IF I GTR M THEN GO TO TOP;                       00108300
IF J GTR U THEN GO TO BOTTOM;                           00108400
IF SORTA[I].[21:27] LEQ SORTA[J].[21:27] THEN GO TO BOTTOM; 00108500
TOP: MERGE[K]:=SORTA[J];                                00108600
J:=J+1;                                                  00108700
IF K:=K+1 LEQ U THEN GO TO AGAIN ELSE GO EXIT;          00108800
BOTTOM: MERGE[K]:=SORTA[I];                              00108900
I:=I+1;                                                  00109000
IF K:=K+1 LEQ U THEN GO TO AGAIN;                       00109100
EXIT: MOVE(U-L+1,MERGE[L],SORTA[L]);                    00109200
END                                                       00109300
END                                                       00109400
END SORT;                                                00109500
*****00109600
PROCEDURE DISKWAIT(I,S,D);                                00109700
VALUE I,S,D; REAL I,S,D; ARRAY A[*]; COMMUNICATE(-8);   00109800
*****00109900
REAL STREAM PROCEDURE DECONV(X); VALUE X;                00110000
BEGIN SI:=LOC X; DI:=LOC DECONV; DS:= 8 DEC; END;       00110100
*****00110200
*****00110300
PROCEDURE SETUSER(CODE); VALUE CODE; REAL CODE; COMMUNICATE(-10); 00110400
REAL STREAM PROCEDURE WHEREIS(X); % GET ADDRESS OF "X"    00110500
BEGIN SI:=X; WHEREIS:=SI; END;                           00110600
*****00110700
REAL PROCEDURE SIZ(A1,A2); VALUE A1,A2; REAL A1,A2; % GET SIZE FROM ADRS00110800
SIZ:=(A2,[33:15]-A1,[33:15])x8+A2.[30:3]-A1.[30:3];     00110900

```

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

```

*****00111000
STREAM PROCEDURE REFORMAT(A2,B1,B4,SEQ,IFLG,SFLG,EADRS); 00111100
*****00111200
VALUE A2,B1,B4,SEQ,IFLG,SFLG,EADRS; 00111300
COMMENT A2 IS STARTING ADDRESS OF STRING IN ARRAY "A", 00111400
B1 IS ADDRESS OF B[0], 00111500
B4 IS ADDRESS OF B[9] IF NOT TYPE DATA, OF B[10] IF TYPE DATA 00111600
SEQ IS OCTAL VALUE OF SEQUENCE NUMBER IF NOT TYPE DATA, 00111700
IFLG IS TRUE IF SEQUENCE NUMBER APPEARS IN INPUT RECORD, 00111800
SFLG IS TRUE IF NOT TYPE DATA FILE, 00111900
EADRS IS ADDRESS OF LAST LEFT ARROW IN USERS INPUT RECORD 00112000
ROUTINE IS USED TO CONVERT INPUT FROM TANK TO STANDARD (80 CHR) FORM; 00112100
BEGIN LABEL L; LOCAL SV,DV; DI:=EADRS; DS:=LIT LEFTARROW; 00112200
DI:=B1; DS:=8LIT" "; SI:=B1; DS:=9WDS; SI:=A2; 00112300
IFLG( % SEQUENCE NUMBER IN RECORD 00112400
63(IF SC=" " THEN SI:=SI+1 ELSE JUMP OUT); 00112500
08(IF SC LSS "0" THEN JUMP OUT;IF SC GTR "9" THEN JUMP OUT;SI:=SI+1);00112600
); 00112700
DI:=B1; 00112800
2(40( 00112900
IF SC=LEFTARROW THEN % CHECK FOR END OF RECORD 00113000
BEGIN 00113100
SV:=SI; SI:=LOC SV; SI:=SI+5; 00113200
DV:=DI; DI:=LOC EADRS; DI:=DI+5; 00113300
IF 3SC=DC THEN JUMP OUT 2 TO L; 00113400
SI:=SV; DI:=DV; 00113500
END; 00113600
DS:=CHR)); 00113700
L: SFLG(SI:=LOC SEQ; DI:=B4; DS:=8DEC); 00113800
END STREAM PROCEDURE REFORMAT; 00113900
*****00114000
BOOLEAN STREAM PROCEDURE FIXCHK(A2,O1,O2,N1,N2,EADRS); VALUE A2,EADRS; 00114100
*****00114200
COMMENT A2 IS STARTING ADDRESS OF STRING IN ARRAY "A", 00114300
O1,O2 ARE START/FINISH ADDRESS OF "OLD STRING", 00114400
N1,N2 ARE START/FINISH ADDRESS OF "NEW STRING". 00114500
EADRS IS ADDRESS OF LAST LEFT ARROW IN USERS INPUT RECORD 00114600
ROUTINE EXAMINES "FIX" SYNTAX AND ESTABLISHES VALUES FOR 00114700
O1,O2,N1 AND N2; 00114800
BEGIN LOCAL SV,D; LABEL EXIT; 00114900
DI:=EADRS; DS:=LIT LEFTARROW; SI:=A2; DI:=LOC D; 00115000
20(IF SC LSS "0" THEN SI:=SI+1 ELSE JUMP OUT); % SCAN TO DIGIT 00115100
IF TOGGLE THEN GO TO EXIT; % NO DIGITS 00115200
08(IF SC LSS "0" THEN JUMP OUT;IF SC GTR "9" THEN JUMP OUT;SI:=SI+1);00115300
20(IF SC EQL " " THEN SI:=SI+1 ELSE JUMP OUT); % SCAN TO CHAR, 00115400
IF TOGGLE THEN GO TO EXIT; % EMPTY FIELD 00115500
DI:=LOC D; DI:=DI+7; DS:=CHR; % SAVE DELIMITER 00115600
IF SC=D THEN GO TO EXIT; % NO STRING 00115700
IF SC = LEFTARROW THEN 00115800
BEGIN 00115900
SV:=SI; SI:=LOC SV; SI:=SI+5; 00116000
DI:=LOC EADRS; DI:=DI+5; 00116100
IF 3SC=DC THEN % LEFT ARROW REPLACEMENT 00116200
BEGIN 00116300
SI:=SV; SI:=SI-1; SV:=SI; SI:=LOC SV; DI:=O1; DS:=WDS; SI:=SV; 00116400
SI:=SI+1; SV:=SI; SI:=LOC SV; DI:=O2; DS:=WDS; 00116500
SI:=LOC SV; DI:=N1; DS:=WDS; SI:=SV; 00116600
SI:=SI+1; SV:=SI; SI:=LOC SV; DI:=N2; DS:=WDS; 00116700
TALLY:=1; FIXCHK:=TALLY; GO TO EXIT; 00116800
END LEFT ARROW REPLACEMENT; 00116900

```

```

SI:=SV; 00117000
END; % IF LEFTARROW 00117100
SV:=SI; SI:=LOC SV; DI:=O1; DS:=WDS; SI:=SV; % START OF OLD STRING 00117200
63( 00117300
IF SC=LEFTARROW THEN % CHECK FOR END OF RECORD 00117400
BEGIN 00117500
SV:=SI; SI:=LOC SV; SI:=SI+5; 00117600
DI:=LOC EADRS; DI:=DI+5; 00117700
IF 3SC=DC THEN JUMP OUT TO EXIT ELSE SI:=SV; 00117800
END; 00117900
IF SC NEQ D THEN SI:=SI+1 ELSE JUMP OUT; 00118000
IF TOGGLE THEN GO TO EXIT; % MISSING 2ND DELIMITER 00118100
SV:=SI; SI:=LOC SV; DI:=O2; DS:=WDS; SI:=SV; %END OLD STRING 00118200
SI:=SI+1; SV:=SI; SI:=LOC SV; DI:=N1; DS:=WDS; SI:=SV; % NEW STRING 00118300
63(IF SC=LEFTARROW THEN 00118400
BEGIN 00118500
SV:=SI; SI:=LOC SV; SI:=SI+5; DI:=LOC EADRS; DI:=DI+5; 00118600
IF 3SC=DC THEN JUMP OUT ELSE SI:=SV; 00118700
END; 00118800
SI:=SI+1); 00118900
SI:=LOC SV; DI:=N2; DS:=WDS; % END NEW STRING 00119000
TALLY:=1; FIXCHK:=TALLY; 00119100
EXIT; 00119200
END STREAM PROCEDURE FIXCHK; 00119300
*****00119400
BOULEAN STREAM PROCEDURE LOCATE(B1,B2,B3,B4,L,O1,SZ0); 00119500
*****00119600
COMMENT B1 IS ADDRESS OF B[0], 00119700
B2 IS ADDRESS OF START OF "EXISTING" STRING, 00119800
B3 IS ADDRESS OF CHARACTER FOLLOWING LAST NON-BLANK CHR. IN RECORD, 00119900
B4 IF ADDRESS OF B[9] IF NOT TYPE DATA, OF B[10] IF TYPE DATA 00120000
L IS RECORD SIZE,(2x)36 CHR. IF NOT TYPE DATA, (2x)40 CHR. IF TYPE DATA. 00120100
ROUTINE MATCHES "OLD STRING" AT ADDRESS O1,LENGTH SZ0 WITH 00120200
CONTENTS OF RECORD FOR "FIX" STATEMENT,AND SETS B2 AND B3 IF MATCHED; 00120300
VALUE B1,B4,L,O1,SZ0; 00120400
BEGIN LOCAL SV,D; LABEL L1,L2,EXIT; 00120500
SI:=O1; DI:=LOC D; DI:=DI+7; DS:=CHR; % SAVE FIRST CHARACTER 00120600
SI:=B1; 00120700
2(L(IF SC=D THEN 00120800
BEGIN 00120900
SV:=SI; DI:=O1; IF SZ0 SC = DC THEN 00121000
BEGIN 00121100
SI:=LOC SV; DI:=B2; DS:=WDS; JUMP OUT 2 TO L1; 00121200
END; 00121300
SI:=SV; 00121400
END; 00121500
SI:=SI+1)); 00121600
GO TO EXIT; 00121700
L1: SI:=B4; SI:=SI-1; 00121800
2(L(IF SC=" " THEN SI:=SI-1 ELSE JUMP OUT 2 TO L2)); 00121900
L2: SI:=SI+1; SV:=SI; SI:=LOC SV; DI:=B3; DS:=WDS; 00122000
TALLY:=1; LOCATE:=TALLY; 00122100
EXIT; 00122200
END STREAM PROCEDURE LOCATE; 00122300
*****00122400
STREAM PROCEDURE EDIT(B2,B4,B5,SZ0,N1,SZN,SFLG); 00122500
*****00122600
COMMENT B2 IS STARTING ADDRES OF "EXISTING" STRING, 00122700
B4 IS ADDRESS OF B[9] IF NOT TYPE DATA, OF B[10] IF TYPE DATA, 00122800
B5 IS ADDRESS OF B[20], 00122900

```

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

```

SZU IS LENGTH OF "EXISTING" STRING,                                00123000
N1 IS STARTING ADDRESS OF REPLACEMENT STRING,                      00123100
SZN IS LENGTH OF REPLACEMENT STRING,                              00123200
SFLG IS TRUE IF NOT TYPE DATA FILE,                              00123300
ROUTINE REPLACES "OLD" STRING WITH "NEW" STRING FOR "FIX";       00123400
VALUE B2,B4,B5,SZO,N1,SZN,SFLG;                                   00123500
BEGIN LOCAL SEQ;                                                  00123600
SFLG(SI:=B4; DI:=LOC SEQ; DS:=WDS); % SAVE SEQUENCE NO.          00123700
DI:=B4; 2(DS:=40LIT" ");                                          00123800
SI:=B2; SI:=SI+SZO; DI:=B5; 2(DS:=40CHR);                         00123900
SI:=N1; DI:=B2; DS:=SZN CHR;                                       00124000
SI:=B5; 2(DS:=40CHR);                                              00124100
SFLG(SI:=LOC SEQ; DI:=B4; DS:=WDS);                                00124200
END STREAM PROCEDURE EDIT;                                         00124300
%*****                                                             00124400
% STREAM PROCEDURE WHICH CONVERTS OCTAL TO BCD                     00124500
%*****                                                             00124600
STREAM PROCEDURE CONCONV(B,C); VALUE B;                             00124700
BEGIN                                                                00124800
DI:=C;                                                                00124900
SI:=LOC B;                                                            00125000
DS:=8 DEC;                                                            00125100
END;                                                                    00125200
%*****                                                             00125300
% STREAM PROCEDURES TO CREATE MESSAGES FOR USERS ERROR MESSAGE FILE 00125400
%*****                                                             00125500
STREAM PROCEDURE ERRORMES(E,ADDR,SEQNO);                             00125600
VALUE ADDR,SEQNO;                                                    00125700
BEGIN                                                                00125800
DI:=E; DS:=8LIT" "; SI:=E; DS:=8WDS;                                00125900
SI:=LOC SEQNO; DS:=8 DEC; DI:=E;                                     00126000
DS:=24 LIT"MAY BE AN ERROR AROUND ";                               00126100
SI:=LOC ADDR; DS:=WDS; DI:=DI-8; DS:=7 FILL;                       00126200
END;                                                                    00126300
STREAM PROCEDURE SEQMES(E,ADDR1,ADDR2,SEQNO);                       00126400
VALUE ADDR1,ADDR2,SEQNO;                                             00126500
BEGIN LOCAL SV;                                                      00126600
DI:=E; DS:=8LIT" "; SI:=E; DS:=8WDS;                                00126700
SI:=LOC SEQNO; DS:=8 DEC; DI:=E;                                     00126800
DS:=17LIT"SEQUENCE ERROR: ";                                       00126900
SI:=LOC ADDR2; DS:=8 DEC; SV:=DI; DI:=DI-8; DS:=7 FILL; DI:=SV;   00127000
DS:=5LIT" - "; SI:=LOC ADDR1; DS:=8 DEC; DI:=DI-8; DS:=7 FILL;    00127100
END;                                                                    00127200
%*****                                                             00127500
STREAM PROCEDURE BLANKLINE(A,SEQN); VALUE SEQN;                     00127600
%*****                                                             00127700
BEGIN                                                                00127800
DI:=A; DS:=8LIT" "; SI:=A; DS:=8WDS; SI:=LOC SEQN; DS:=8DEC;      00127900
END;                                                                    00128000
DEFINE IOMASK = 3"2000000000"#;                                     00128100
%                                                                      00128200
%*****                                                             00128300
% STREAM PROCEDURE TO SCAN USERS INPUT FOR FILENAMES              00128400
%*****                                                             00128500
%                                                                      00128600
REAL STREAM PROCEDURE SCN(A,WD);                                     00128700
BEGIN                                                                00128800
LABEL XIT,B;                                                         00128900
SI:=A; DI:=A; DI:=DI+23; DS:=LIT" ";                               00129000
B: IF SC=" " THEN BEGIN SI:=SI+1; GO B; END;                         00129100

```

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=ALPHA THEN                                00129200
BEGIN                                           00129300
IF SC GEQ "0" THEN BEGIN %BEGINS WITH A NUMBER 00129400
      TALLY:=2; GO XIT;                          00129500
      END;                                       00129600
DI:=WD; DS:=LIT"0";                             00129700
6(IF SC=ALPHA THEN DS:=CHR ELSE DS:=LIT" ");    00129800
DS:=LIT" ";                                       00129900
END                                             00130000
      ELSE TALLY:=1; %BEGINS WITH BAD CHAR      00130100
XIT: SCN:=TALLY;                                00130200
      END;                                       00130300
*****                                         00130400
PROCEDURE EXPAND(A,RSIZE); VALUE RSIZE;        00130500
*****                                         00130600
REAL RSIZE; ARRAY A(0);                        00130700
BEGIN                                           00130800
DEFINE FTC:=SI:=SI+3; DI:=DI+5; SKIP 3DB;      00130900
15(IF SB THEN DS:=SET ELSE DS:=RESET; SKIP SB)# 00131000
DEFINE CTF:=SI:=SI+5; SKIP 3SB; DI:=DI+3;      00131100
15(IF SB THEN DS:=SET ELSE DS:=RESET; SKIP SB)# 00131200
DEFINE CTC:=SI:=SI+5; SKIP 3SB; DI:=DI+5; SKIP 3DB; 00131300
15(IF SB THEN DS:=SET ELSE DS:=RESET; SKIP SB)# 00131400
SAVE ARRAY B(0:RSIZE);                          00131500
STREAM PROCEDURE CHANGEVECTORS(A,B);           00131600
  BEGIN LOCAL MOMA,MOMB,BS,OLD,SV; LABEL L,EXIT; 00131700
  DI:=LOC BS; DS:=8LIT"BBBBBBBB"; % STACK BOTTOM MARKER 00131800
  SI:=LOC A; DI:=LOC OLD; DS:=WDS; % OLD DATA DESCRIPTOR 00131900
  SI:=LOC A; DI:=LOC MOMA; FTC;                 00132000
  SI:=LOC B; DI:=LOC MOMB; CTF;                 00132100
  SI:=LOC MOMB; DI:=LOC A; CTF;                 00132200
  SI:=LOC MOMA; DI:=LOC B; CTF;                 00132300
  SI:=LOC B; DI:=MOMA; DS:=WDS;                 00132400
  SI:=LOC A; DI:=MOMB; DS:=WDS;                 00132500
  DI:=LOC OLD; DI:=DI-8;                         00132600
L: DI:=DI-8; SV:=DI; SI:=LOC BS; IF SC=DC THEN GO TO EXIT; 00132700
  DI:=SV; SI:=LOC OLD; IF BSC=DC THEN           00132800
  BEGIN                                         00132900
    DI:=SV; SI:=MOMA; DS:=WDS; % REPLACE DATA DESCRIPTOR 00133000
  END;                                         00133100
  DI:=SV; GO TO L;                             00133200
EXIT:                                          00133300
END STREAM PROCEDURE CHANGEVECTORS;           00133400
CHANGEVECTORS(A,B);                           00133500
END PROCEDURE EXPAND;                          00133600
%                                               00133700
*****                                         00133800
% STREAM PROCEDURES FOR USER CONVERSATION     00133900
*****                                         00134000
%                                               00134100
STREAM PROCEDURE MESS(A);                      00134200
BEGIN                                           00134300
DI:=A; DS:=8LIT" "; SI:=A; DS:=8WDS; DI:=A;    00134400
DS:=45LIT"AARRRGH ...TO MUCH MESSED UP ...SURRY, BYE."; 00134500
END;                                           00134600
STREAM PROCEDURE S1(A);                        00134700
BEGIN                                           00134800
DI:=A; DS:=8LIT" "; SI:=A; DS:=8WDS; DI:=A;    00134900
DS:=38LIT"AN ERROR HAS OCCURRED IN THE WORK FILE"; 00135000
END;                                           00135100

```

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

	STREAM PROCEDURE S2(A);	00135200
	BEGIN	00135300
	DI:=A; DS:=8LIT" "; SI:=A; DS:=8WDS; DI:=A;	00135400
1	DS:=41LIT"I WILL TRY TO RECOVER AS MUCH AS POSSIBLE";	00135500
2	END;	00135600
3	STREAM PROCEDURE S3(A);	00135700
4	BEGIN	00135800
5	DI:=A; DS:=8LIT" "; SI:=A; DS:=8WDS; DI:=A;	00135900
6	DS:=51LIT"PLEASE GIVE ME A NAME FOR THE ERROR MESSAGE FILE...";	00136000
7	END;	00136100
8	STREAM PROCEDURE S4(A,P);	00136200
9	BEGIN	00136300
10	DI:=A; DS:=8LIT" "; SI:=A; DS:=8WDS; DI:=A;	00136400
11	DS:=25LIT"AN ERROR HAS OCCURRED IN ";	00136500
12	SI:=P; SI:=SI+9; 7(IF SC=ALPHA THEN DS:=CHR ELSE SI:=SI+1);	00136550
13	END;	00136600
14	STREAM PROCEDURE S5(A);	00136700
15	BEGIN	00136800
16	DI:=A; DS:=8LIT" "; SI:=A; DS:=8WDS; DI:=A;	00136900
17	DS:=45LIT"NOW GIVE ME A NAME FOR THE NEW SOURCE FILE...";	00137000
18	END;	00137100
19	STREAM PROCEDURE S6(A);	00137200
20	BEGIN	00137300
21	DI:=A; DS:=8LIT" "; SI:=A; DS:=8WDS; DI:=A;	00137400
22	DS:=35LIT"THAT IS NOT A NICE NAME...TRY AGAIN";	00137500
23	END;	00137600
24	STREAM PROCEDURE S7(A);	00137700
25	BEGIN	00137800
26	DI:=A; DS:=8LIT" "; SI:=A; DS:=8WDS; DI:=A;	00137900
27	DS:=47LIT"FILENAME MAY NOT START WITH DIGITS...TRY AGAIN";	00138000
28	END;	00138100
29	STREAM PROCEDURE S8(A);	00138200
30	BEGIN	00138300
31	DI:=A; DS:=8LIT" "; SI:=A; DS:=8WDS; DI:=A;	00138400
32	DS:=44LIT"PLEASE GIVE ME A NAME FOR A RECOVERY FILE...";	00138500
33	END;	00138600
34	STREAM PROCEDURE S9(A);	00138700
35	BEGIN	00138800
36	DI:=A; DS:=8LIT" "; SI:=A; DS:=8WDS; DI:=A;	00138900
37	DS:=31LIT"DUPLICATE FILE NAME...TRY AGAIN";	00139000
38	END;	00139100
39	STREAM PROCEDURE S10(A);	00139200
40	BEGIN	00139300
41	DI:=A; DS:=8LIT" "; SI:=A; DS:=8WDS; DI:=A;	00139400
42	DS:=50LIT"I WILL TRY TO RECOVER THE CHANGES AND/OR ADDITIONS";	00139500
43	END;	00139600
44	STREAM PROCEDURE S11(A);	00139700
45	BEGIN	00139800
46	DI:=A; DS:=8LIT" "; SI:=A; DS:=8WDS; DI:=A;	00139900
47	DS:=20LIT"NO CHANGES RECOVERED";	00140000
48	END;	00140100
49	STREAM PROCEDURE S12(A);	00140200
50	BEGIN	00140300
51	DI:=A; DS:=8LIT" "; SI:=A; DS:=8WDS; DI:=A;	00140400
52	DS:=18LIT"NO ERRORS RECORDED";	00140500
53	END;	00140600
54	STREAM PROCEDURE S14(A,NAME); VALUE NAME;	00141200
55	BEGIN	00141300
56	DI:=A; DS:=8LIT" "; SI:=A; DS:=8WDS;	00141400
57	DI:=A; DS:=5LIT"FILE "; DS:=LIT"";	00141500


```

SI:=LOC NAME; SI:=SI+1;                                00141550
DS:=7 CHR; DS:=LIT"";                                  00141600
DS:=33LIT" IS PROTECTED, OR NOT ON THE DISK";          00141650
END;                                                    00141700
STREAM PROCEDURE S15(A);                                00141800
BEGIN                                                  00141900
DI:=A; DS:=8LIT" "; SI:=A; DS:=8WDS; DI:=A;          00142000
DS:=27LIT"RECOVERY ATTEMPT COMPLETED.";             00142100
END; % CONVERSATION ROUTINES                          00142200
%                                                      00142300
%*****                                                00142400
% ROUTINE FOR DIALOG WITH USER TO                    00142500
% DETERMINE WHICH FILES THE USER WOULD HAVE YOU     00142600
% WRITE THE RECOVERY DATA ON                        00142700
%*****                                                00142800
%                                                      00142900
PROCEDURE DIALOG(NUM,WORD); VALUE NUM; INTEGER NUM;   00143000
ARRAY WORD[0];                                       00143100
BEGIN                                                  00143200
FILE DUMMY DISK SERIAL (2,10,30);                    00143300
LABEL UP,UP2,DOWN; ARRAY A[0:8],FN[0:2];            00143400
REAL TYPE;                                           00143500
IF NUM=1 THEN BEGIN %JUST WANT ERROR FILE           00143600
S1(A); WRITE(TWX,9,A[*]); END;                       00143700
IF NUM=2 THEN BEGIN %WANT BOTH ERROR & NEW SOURCE FILE 00143800
S4(A,P); WRITE(TWX,9,A[*]); END;                     00143900
IF NUM=3 THEN BEGIN %JUST WANT RECOVERY SOURCE FILE 00144000
S1(A); WRITE(TWX,9,A[*]);                             00144100
S10(A); WRITE(TWX,9,A[*]);                            00144200
S8(A); WRITE(TWX,9,A[*]); GO UP2; END;                00144300
S2(A); WRITE (TWX,9,A[*]);                             00144400
S3(A); WRITE (TWX,9,A[*]);                             00144500
UP: READ (TWX,3,FN[*]);                                00144600
TYPE:=SCN(FN,WORD[0]); %ERROR FILE RETURNED IN WORD[0] 00144700
IF TYPE = 1 THEN BEGIN %BEGINS NON-ALPHA            00144800
S6(A); WRITE (TWX,9,A[*]); GO UP;                     00144900
END;                                                    00145000
IF TYPE = 2 THEN BEGIN %BEGINS WITH NUMBER          00145100
S7(A); WRITE (TWX,9,A[*]); GO UP;                     00145200
END;                                                    00145300
FILL DUMMY WITH WORD[0],USER;                          00145400
SEARCH(DUMMY,A[*]);                                    00145500
IF (A[0] GEQ 0 OR WORD[0]=WORD[1]) THEN BEGIN %DUPLICATE FILE 00145600
S9(A); WRITE(TWX,9,A[*]); GO UP; END;                 00145700
IF NUM=1 THEN GO DOWN ELSE BEGIN                     00145800
S5(A); WRITE(TWX,9,A[*]); END;                         00145900
UP2: READ (TWX,3,FN[*]);                               00146000
TYPE:=SCN(FN,WORD[1]); %SOURCE FILE RETURNED IN WORD[1] 00146100
IF TYPE = 1 THEN BEGIN %BEGINS NON-ALPHA            00146200
S6(A); WRITE (TWX,9,A[*]); GO UP2;                   00146300
END;                                                    00146400
IF TYPE = 2 THEN BEGIN %BEGINS WITH NUMBER          00146500
S7(A); WRITE (TWX,9,A[*]); GO UP2;                   00146600
END;                                                    00146700
FILL DUMMY WITH WORD[1],USER;                          00146800
SEARCH(DUMMY,A[*]);                                    00146900
IF (A[0] GEQ 0 OR WORD[0]=WORD[1]) THEN BEGIN %DUPLICATE FILE 00147000
S9(A); WRITE(TWX,9,A[*]); GO UP2; END;                 00147100
DOWN:                                                  00147200
END; % SCRATCH FILE DETERMINATOR                     00147300

```

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

```

%*****00147400
REAL STREAM PROCEDURE INPCONV(A);                                00147500
  BEGIN SI:=A; DI:=LCC INPCONV; DS:=B OCT; END;                    00147600
%                                                                    00147700
%*****00147800
%*****00147900
%*****00148000
% PROCEDURE TO HANDLE A "1T" PARITY ERROR                          00148100
%*****00148200
%*****00148300
%                                                                    00148400
%INPUT: SOURCE FILE                                              00148500
%OUTPUT: NEW "1T" FILE CORRESPONDING TO SOURCE                    00148600
% ENTERED IF ERROR OCCURS IN "1T" FILE...                          00148700
% IT WILL ATTEMPT TO REBUILD A GOOD "1T" FILE FROM SOURCE        00148800
% IF PARITY OCCURS IN SOURCE, HAND1T SET TRUE, CONTROL WILL      00148900
% BE TRANSFERED TO "1S" HANDLER UPON EXIT.                        00149000
BOOLEAN PROCEDURE HAND1T(A); ARRAY A[0];                          00149100
  BEGIN                                                            00149200
    FILE SOUR2 DISK SERIAL (2,10,30);                               00149300
    FILE TFILE DISK SERIAL [20:600] (2,30,300,SAVE 7);             00149400
    LABEL OKOUT,EOF,SP,SP2; ARRAY B[0:30];                          00149500
    INTEGER T,PREVSN;                                                00149600
    FILL SOUR2 WITH SRCNAME,P[6];                                    00149700
    FILL TFILE WITH " "&"1T"[6:30:12]&LINE[18:30:18],P[6];         00149800
    B[0]:=0; %ZERO FIRST WORD OF "1T" FILE                          00149900
    PREVSN:=-1;                                                      00150000
    T:=0;                                                            00150100
    DO BEGIN %BASIC READ SOURCE, WRITE "1T" LOUF                    00150200
      READ(SOUR2,10,A[*])[EOF:SP]; %ON EOF SET MARKER AND EXIT     00150300
      IF (T:=T+1) GTR 29 THEN                                        00150400
        BEGIN                                                       00150500
          WRITE(TFILE,30,B[*]);                                     00150600
          T:=0; END;                                               00150700
          B[T]:=INPCONV(A[9]); %CONVERT SEQ NO TO OCTAL              00150800
          IF B[T] LEQ PREVSN THEN GO SP2 ELSE PREVSN:=B[T];        00150900
        END UNTIL T=30;                                             00151000
      EOF: IF T LSS 29 THEN B[T+1]:=100000000                      00151100
          ELSE BEGIN                                                00151200
            WRITE(TFILE,30,B[*]);                                   00151300
            B[0]:=100000000;                                         00151400
            END;                                                    00151500
            WRITE(TFILE,30,B[*]);                                   00151600
            LOCK(TFILE,*);                                           00151700
            GO OKOUT;                                                00151900
          SP: READ(SOUR2,10,A[*]);                                    00151950
          SP2: HAND1T:=TRUE; LOCK(TFILE,*);                          00152000
          OKOUT:                                                     00152100
            END; % "1T" HANDLER                                     00152200
          %                                                            00152300
          %*****00152400
          %*****00152500
          %*****00152600
          %*****00152700
          % PROCEDURE TO HANDLE "1S" OR SOURCE PARITY ERRORS        00152800
          %*****00152800
          %*****00152900
          %                                                            00153000
          %INPUT: SOURCE FILE                                        00153100
          %OUTPUT: NEW SOURCE FILE                                  00153200
          % CORRESPONDING ERROR MESSAGE FILE                          00153300

```

Data Documents, Inc.

```

% POSSIBLE NEW "1T" FILE                                00153400
% ENTERED IF ERROR OCCURS IN SOURCE OR "1S" FILE        00153500
% IT WILL IGNORE ERRORS IN THE SOURCE FILE, CREATING A NEW 00153600
% SOURCE FILE AND NOTING WHERE ERRORS MAY HAVE OCCURRED 00153700
% IF THE SOURCE WAS BOTH THE WORKFILE AND SEQUENTIAL A 00153800
% CORRESPONDING "1T" FILE WILL BE BUILT.                00153900
BOOLEAN PROCEDURE HAND1S(A,B,P);                          00154000
ARRAY P[0]; ARRAY A[0]; ARRAY B[0];                      00154100
BEGIN DWN BOOLEAN BHB;                                    00154200
FILE SOUR2 DISK SERIAL (2,0,0);                           00154300
FILE TFILE DISK SERIAL [20:600] (2,30,300,SAVE 7);       00154400
FILE USERER DISK SERIAL [20:600] (2,10,300,SAVE 7);       00154500
FILE USERSQ DISK SERIAL [20:600] (2,10,300,SAVE 7);       00154600
INTEGER PP,N,T,I,J,L,SEQNO,RECS,PREVS,NOR;                00154700
BOOLEAN NOR,ERRIDG,USERSQDQ;                              00154800
ARRAY ERRMES[0:9];                                        00154900
LABEL END1S,L1,L2,L4,L5,SP1,OUTS,ABORT,LSPC1,LSPC2;       00155000
FILL SOUR2 WITH SRCNAME,P[6];                             00155100
FILL TFILE WITH " "&"1T"[6:36:12]&LINE[18:30:18],P[6];   00155200
FILL USERER WITH WORD[0],P[6];                            00155300
FILL USERSQ WITH DNAME,P[6];                              00155400
SDATA[0]:=-1; GETHEADER(SDATA,P[1],P[6],P[6]);           00155500
IF SDATA[0] LEQ 0 THEN BEGIN %NON-EXISTENT OR PROTECTED FILE
    S14(ERRMES,P[1]);                                     00155600
    WRITE(TWX,6,ERRMES[*]);                             00155700
    GO OUTR;                                             00155800
    END;                                                 00155900
IF SDATA[0].[42:6]=0 THEN GO OUTRMOST;                    00156000
IF SDATA[0].[15:15]=0 THEN SDATA[0].[15:15]:=SDATA[0].[42:6]*30; 00156010
IF SDATA[0].[30:12]=0 THEN SDATA[0].[30:12]:=1;         00156020
IF SDATA[0].[1:14]=0 THEN SDATA[0].[1:14]:=SDATA[0].[15:15] DIV 00156030
    SDATA[0].[30:12];
NOR:=SDATA[7]+1; %NUMBER OF RECORDS IN SOURCE            00156040
IF NOT WRKFILE THEN BEGIN                                00156050
    RESETSIZE(USERSQ,SDATA[0].[1:14],SDATA[0].[15:15],20,600); 00156100
    IF (SDATA[0].[15:15] MOD 30)=0 THEN SDATA[0].[15:15]:=30; 00156200
    RESETSIZE(SOUR2,SDATA[0].[1:14],SDATA[0].[15:15],0,0); 00156250
    RECS:=SDATA[0].[1:14];                               00156300
    EXPAND(A,RECS); EXPAND(ERRMES,RECS);                 00156350
    % IF TYPE IS NOT DATA AND RECORD SIZE IS 10 THEN SEQUENTIAL 00156400
    SFLG:=SDATA[4].[36:6] NEQ 8 AND (SDATA[0].[1:14]=10 OR SDATA[7]=-1); 00156500
    END;                                                 00156600
ELSE                                                     00156700
    BEGIN                                               00156800
    SFLG:=P[5].[34:6] NEQ 8;                              00156900
    RESETSIZE(SOUR2,10,30,0,0);                          00157000
    RECS:=10;                                             00157100
    END;                                                 00157150
IF BHB THEN GO ABORT; %IF ROUTINE HAS BEEN ENTERED BEFORE, ABORT 00157200
BHB:=TRUE;                                               00157300
N:=1;                                                    00157400
B[0]:=0; %ZERO FIRST WORD OF "1T" FILE                  00157500
PREVSN:=-1;                                              00157600
%IF NO RECORDS IN SOURCE CREATE "1T" AND EXIT           00157800
IF(WRKFILE AND SFLG AND NOR=0) THEN GO L4;               00157900
L1: IF (WRKFILE AND SFLG) THEN T:=T+1;                  00158000
L2: READ(SOUR2,RECS,A[*J])[1:SP1]; %BASIC READ SOURCE, WRITE 00158100
    PP:=INPCONV(A[9]); %NEW SOURCE LOOP                  00158200
    IF (WRKFILE AND SFLG) THEN BEGIN                     00158300
    IF PP LEQ PREVSN THEN BEGIN %IS SEQ. NO. OUT OF ORDER 00158400

```

Data Documents/11.c

	PP:=PREVSN+1; %SET IT TO PREVIOUS SEQ. NO. + 1	00158700
	SEQNO:=SEQNO+2;	00158800
	SEQMES(ERRMES,PP,PREVSN,SEQNO); %NOTIFY USER OF CHANGE	00158900
1	WRITE(USERER,RECS,ERRMES[*]);	00159000
2	ERRTOG:=TRUE;	00159100
3	CONCONV(PP,A[9]); %PUT NEW SEQ. NO. IN SEQ. FIELD	00159200
4	END;	00159300
5	B[T]:=PP;	00159400
6	END;	00159500
7	PREVSN:=PP;	00159600
8	WRITE(USERSO,RECS,A[*]);	00159700
9	USERSOTOG:=TRUE;	00159800
10	N:=N+1;	00159900
11	IF N GTR NOR THEN GO L4;	00160000
12	IF (NOT (WRKFILE AND SFLG)) THEN GO LSPC1;	00160100
13	T:=T+1;	00160200
14	IF T LEQ 29 THEN GO L2;	00160300
15	L5: IF (WRKFILE AND SFLG) THEN WRITE(TFILE,30,B[*]);	00160400
16	LSPC1: IF NOR THEN GO OUTS ELSE BEGIN T:=-1; GO L1; END;	00160500
17	L4: NOR:=TRUE;	00160600
18	IF (WRKFILE AND SFLG) THEN %STORE EOF MARKER	00160700
19	IF T LSS 29 THEN B[T+1]:=100000000	00160800
20	ELSE BEGIN	00160900
21	WRITE(TFILE,30,B[*]);	00161000
22	B[0]:=100000000;	00161100
23	END;	00161200
24	GO L5;	00161300
25	SP1: READ(SOUR2,RECS,A[*]); %SOURCE PARITY HAS OCCURRED	00161400
26	PP:=PP+1;	00162200
27	PREVSN:=PP;	00162300
28	IF (WRKFILE AND SFLG) THEN B[T]:=PP;	00162400
29	CONCONV(PP,A[9]); %CONVERT OCTAL SEQ NO TO BCD	00162500
30	MOVE(10,A[0],ERRMES[0]); %CORRECTED SEQ NO	00162600
31	WRITE(USERSO,RECS,ERRMES[*]); %WRITE POSSIBLY BAD RECORD	00162700
32	USERSOTOG:=TRUE; %TO NEW SOURCE FILE	00162800
33	SEQNO:=SEQNO+2;	00162900
34	ERRMES(ERRMES,A[9],SEQNO);	00163000
35	WRITE(USERER,RECS,ERRMES[*]); %WRITE CORRESPONDING ERROR NOTE	00163100
36	ERRTOG:=TRUE;	00163200
37	IF (NOT (WRKFILE AND SFLG)) THEN GO LSPC2;	00163400
38	T:=T+1;	00163500
39	IF T GTR 29 THEN BEGIN WRITE(TFILE,30,B[*]); T:=0; END;	00163600
40	LSPC2: N:=N+1;	00163700
41	IF N GTR NOR THEN GO L4 ELSE GO L2;	00163800
42	ABORT: HAND1S:=TRUE; %ROUTINE HAS BEEN ENTERED BEFORE	00164400
43	GO END1S;	00164600
44	OUTS: IF (NOT ERRTOG) THEN BEGIN S12(ERRMES); %NO ERRORS TO REPORT	00164700
45	WRITE(USERER,RECS,ERRMES[*]);	00164800
46	ERRTOG:=TRUE;	00164900
47	END;	00165000
48	CLOSE(SOUR2);	00165500
49	IF (WRKFILE AND SFLG) THEN BEGIN	00165600
50	SRCEMAME:=" "&"1S"[6:36:12]&LINE[18:30:18];	00165700
51	COMMON.[3:1]:=1; %WORKFILE NAME CHANGED	00165800
52	END;	00165900
53	IF USERSOTOG THEN MOVETYPE(SDATA[4],[36:6],USERSO);	00166000
54	LOCK(USERSO,*);	00166100
55	IF ERRTOG THEN MOVETYPE(10,USERER);	00166200
56	LOCK(USERER,*);	00166300
57	IF (WRKFILE AND SFLG) THEN LOCK(TFILE,*);	00166400

```

END1S:
END; % "1S" AND SOURCE PARITY HANDLER
%
%*****
%*****
%*****
% ROUTINE TO HANDLE "1P" PARITY ERRORS
%*****
%*****
%
%INPUT: "1P" FILE
% "1T" FILE
% "1S" FILE
%OUTPUT: CHANGE AND/OR ADDITIONS FILE
% ENTERED IF ERROR NOTED IN "1P" FILE
% ACTION TAKEN,
% CHANGES SORTED FOR MERGE WITH "1T" FILE...
% IF ERROR NOTED IN "1T", "1T" HANDLER CALLED
% IF ERROR NOTED IN "1S", "1S" HANDLER CALLED
% ...THEN "1P" HANDLER REINITIALIZED WITH APPROPRIATE
% NEW FILES TO WORK WITH
% USER CHANGES AND/OR ADDITIONS ARE WRITTEN TO RECOVERY FILE.
PRUCEDURE HAND1P(SORTA,MAXCNT);
ARRAY SORTA(0);
INTEGER MAXCNT;
BEGIN INTEGER CC,N1,N2; LABEL UP,ASN,PPAR,PFLG,
COMPARE,ENDMERGE,NEXTIP,PCYCLE,SKIP,STARTMERGE,
TLOOP,WRITENEW,EOF,OTPAR,SOUPAR,UTPAR2;
REAL CODE,MAX,NCHRS,PCT,PREV,PPTR,PSEQ,PWORD,SZN,SZO,
TANKADRS,TCT,TPTR,TSEQ;
BOOLEAN BFULL,EQLTOG,PGMTOG;
FILE PFILE DISK SERIAL (2,30);
FILE SOUR1 DISK RANDOM (2,10,30); % RANDOM SOURCE FILE
FILE OLDTAB DISK SERIAL (2,30,300); % OLD "1T" FILE
FILE PATCH DISK SERIAL [20:600] (2,10,300,SAVE 7); % SCRATCH PAD FILE
B4:=WHEREIS(B[10-REAL(SFLG)]); B5:=WHEREIS(B[20]);
RECSZ:=40-4*REAL(SFLG); % 1/2 RECORD LENGTH
EXPAND(SORTA,MAXCNT); EXPAND(MERGE,MAXCNT);
FLAG:=PFLG; %SET TO CHECK FOR FLAG BIT ERROR
CC:=N2:=0;
FILL PFILE WITH " "&"1P"[6:36:12]&LINE[18:30:18],P[6];
UP: N1:=0;
READ(PFILE,30,A[*])[:PPAR]; %IGNORE PARITY ON "1P" FILE
GO ASN;
PPAR:
ASN: IF CC LSS MAXCNT THEN %CHECK AGAINST NO OF CHANGES
%TO WORKFILE
IF A[N1] NEQ 10000000 THEN %CHECK FOR EOF MARKER IN "1P"
BEGIN
SORTA[N1+N2]:=A[N1]; %IF FLAG BIT ON STORE "0"
PFLG: N1:=N1+1;
CC:=CC+1;
IF N1 GTR 29 THEN
BEGIN
N2:=N2+30;
GO UP;
END
ELSE
GO ASN;
END;
END;

```

```

00166600
00166700
00166800
00166900
00167000
00167100
00167200
00167300
00167400
00167500
00167600
00167700
00167800
00167900
00168000
00168100
00168200
00168300
00168400
00168500
00168600
00168700
00168800
00168900
00169000
00169100
00169200
00169300
00169400
00169500
00169600
00169700
00169800
00169900
00170000
00170100
00170200
00170300
00170400
00170500
00170600
00170700
00170800
00170900
00171000
00171100
00171200
00171300
00171400
00171500
00171600
00171700
00171800
00171900
00172000
00172100
00172200
00172300
00172400
00172500

```

SORTA[N2:=N2+N1]:=100000000; *INSERT EOF MARKER

00172600

SORT(0,N2);

00172700

EXPAND(MERGE,1);

00172800

%.....

00172900

STARTMERGE:

00173000

%.....

00173100

FILL SOUR1 WITH SRCENAME,P[6];

00173200

FILL OLDTAB WITH " "&"1T"[6:36:12]&LINE[18:30:18],P[6];

00173300

FILL PATCH WITH VARNAME,P[6]; * FOR USER SPECIFIED RECOVERY FILE

00173400

PSEQ:=(PWORD:=SORTA[0]).[21:27]; * 1ST SEQ. NUMBER IN "1P" FILE

00173500

CODE:=PWORD.[1:2]; PPTR:=0;

00173600

PCT:=TCT:=-1;

00173700

IF SFLG THEN * NOT TYPE DATA FILE

00173800

BEGIN

00173900

READ(OLDTAB,30,T[*])LUTPAR:QTPAR]; TPTR:=0;

00174000

END

00174100

ELSE

00174200

BEGIN

00174300

SEARCH(SOUR1,SDATA[*]);

00174400

MAX:=SDATA[5]+1;

00174500

TSEQ := 0;

00174600

END;

00174700

%.....

00174800

TLOOP:

00174900

%.....

00175000

IF SFLG THEN * SEQUENCED FILE

00175100

BEGIN

00175200

IF TPTR:=TPTR+1 GTR 29 THEN * SEGMENT IS EXHAUSTED

00175300

BEGIN

00175400

READ(OLDTAB,30,T[*])[QTPAR:QTPAR]; * GET NEW SEGMENT

00175500

TPTR:=0;

00175600

END;

00175700

FLAG:=QTPAR2;

00175800

TSEQ:=T[TPTR].[21:27]; TCT:=TCT+1; * OLD SEQ. NO. AND LOCATION

00175900

IF TSEQ=EOFMARK THEN CLOSE(OLDTAB);

00176000

END

00176100

ELSE

00176200

BEGIN * NON-SEQUENCED TYPE FILE

00176300

IF TSEQ := (TCT:=TSEQ) + 1 GTR MAX THEN

00176400

BEGIN

00176500

TSEQ := EOFMARK; CLOSE(OLDTAB);

00176600

TCT := MAX;

00176700

END;

00176800

END;

00176900

%.....

00177000

COMPARE:

00177100

%.....

00177200

IF PWORD=0 THEN GO NEXTP;

00177300

IF TSEQ LSS PSEQ THEN * NO CHANGE TO THIS RECORD

00177400

BEGIN

00177500

EQLTOG:=TRUE; SEQ:=TSEQ; GO TO WRITENEW;

00177600

END;

00177700

EQLTOG := TSEQ = PSEQ; * NEW RECORD SEQ. MATCHES OLD

00177800

%.....

00177900

PCYCLE:

00178000

%.....

00178100

IF PSEQ=EOFMARK THEN GO TO ENDMERGE;

00178200

IF CODE=1 THEN * DELETE CODE

00178300

BEGIN

00178400

BFULL:=FALSE;

00178500

	PREV:=PSEQ; % SAVE CURRENT PSEQ NUMBER	00178600
	PPTR:=PPTR+1; % GET THE NEXT PSEQ NUMBER	00178700
	PSEQ:=(PWORD:=SORTALPPTR),[21:27];	00178800
1	CODE:=PWORD.[1:2];	00178900
2	IF PSEQ=PREV AND CODE.[47:1]=0 THEN GO PCYCLE; % NEW RECORD FOLLOWS	00179000
3	IF EQLTOG THEN % "OLD" RECORD IS DELETED	00179100
4	BEGIN	00179200
5	BLANKLINE(A,PREV);	00179300
6	IF SFLG THEN BEGIN %WRITE BLANK RECORD IF TYPE SEQ	00179400
7	WRITE(PATCH,10,A[*]);	00179500
8	PGMTOG:=TRUE;	00179600
9	END;	00179700
10	GO TO TLOOP; % ADVANCE "OLDTAB" POINTER	00179800
11	END; % IF EQLTOG	00179900
12	GO TO COMPARE; % IF NOT EQLTOG	00180000
13	END; % IF CODE = 1	00180100
14	IF CODE=3 THEN % FIX CODE, CHECK FOR RECORD FIRST	00180200
15	BEGIN	00180300
16	IF NOT (EQLTOG OR BFULL) THEN % NO SUCH RECORD	00180400
17	BEGIN	00180500
18	GO TO NEXTP;	00180600
19	END;	00180700
20	END;	00180800
21	TANKADRS:=BASE[PWORD.[4:4]] + PWORD.[8:8]; % TANK ROW ADDRESS	00180900
22	DISKWAIT(1,A,30,TANKADRS); % GET INPUT FROM TANK FILE	00181000
23	A2 := A1 + PWORD.[16:5]; % STARTING ADDRESS OF INPUT	00181100
24	NCHRS := SIZEOFINPUT(A2)-1; % POSITION OF LAST CHARACTER IN USERS INPUT	00181200
25	EADRS := (A2 + NCHRS.[40:5]) & NCHRS[30:45:3]; % END OF RCRD ADDRESS	00181300
26	IF CODE NEQ 3 THEN % NOT A FIX COMMAND	00181400
27	BEGIN	00181500
28	REFORMAT(A2,B1,B4,PSEQ,(PWORD.[1:2]=0),SFLG,EADRS);	00181600
29	BFULL:=TRUE; % NEW RECORD IN "B"	00181700
30	GO TO NEXTP;	00181800
31	END;	00181900
32	IF NOT FIXCHK(A2,01,02,N1,N2,EADRS) THEN % "FIX" SYNTAX ERROR	00182000
33	BEGIN	00182100
34	GO TO NEXTP;	00182200
35	END;	00182300
36	IF N2.[33:15] GTR A4 THEN % SHOULD NOT BE HERE	00182400
37	BEGIN	00182500
38	GO TO NEXTP;	00182600
39	END;	00182700
40	IF NOT BFULL THEN % GET RECORD TO BE "FIXED"	00182800
41	BEGIN	00182900
42	READ(SOUR1[TCT],10,B[*])[:SOUPAR]; % PUT IN "B"	00183000
43	BFULL:=TRUE;	00183100
44	END;	00183200
45	IF NOT LOCATE(B1,B2,B3,B4,RECSZ,01,SZO:=SIZ(01,02)) THEN	00183300
46	BEGIN	00183400
47	GO TO NEXTP;	00183500
48	END;	00183600
49	IF SIZ(B3,B4) LSS (SZN:=SIZ(N1,N2))-SZO THEN	00183700
50	BEGIN	00183800
51	GO TO NEXTP;	00183900
52	END;	00184000
53	EDIT(B2,B4,B5,SZO,N1,SZN,SFLG); % REPLACE OLD STRING WITH NEW STRING	00184100
54	%.....	00184200
55	NEXTP: % LOOK AT NEXT "1P" WORD	00184300
56	%.....	00184400
57	PREV:=PSEQ; % SAVE CURRENT "1P" SEQUENCE NUMBER	00184500

Data Documents/Inc.

```

PPTR:=PPTR+1; % CHECK NEXT "1P" WORD                                00184600
PSEQ:=(PWORD:=SORTA[PPTR]).[21:27]; % NEXT "1P" WD.                 00184700
IF PWORD=0 THEN GO NEXTP;                                           00184800
CODE:=PWORD.[1:2];                                                 00184900
IF PREV=PSEQ THEN GO TO PCYCLE; % DUPLICATE, MORE TO BE DONE       00185000
IF NOT BFULL THEN GO TO COMPARE; % IGNORE LAST INPUT               00185100
SEQ:=PREV; % USING THIS SEQ. NUMBER                                00185200
WRITE(PATCH,10,B[*]); % MOVE RECORD TO PATCH FILE AND COUNT       00185300
PGMTOG:=TRUE;                                                       00185400
%.....                                                             00185500
WRITENEW;                                                            00185600
%.....                                                             00185700
BFULL:=FALSE;                                                       00185800
IF EQLTOG THEN GO TO TLOOP ELSE GO TO COMPARE;                     00185900
%.....                                                             00186000
ENDMERGE;                                                            00186100
%.....                                                             00186200
IF (NOT PGMTOG) THEN BEGIN %NO CORRECTIONS WRITTEN                 00186300
  S11(B); WRITE(PATCH,10,B[*]);                                     00186400
  PGMTOG:=TRUE;                                                  00186500
END;                                                                00186600
IF PGMTOG THEN MOVETYPE(P[5],[34:6],PATCH);                       00186700
LOCK(PATCH,*);                                                     00186800
GO OUTR;                                                            00187000
%HERE IF "1T" ERROR NOTED DURING "1P" HANDLING                    00187100
OTPAR: READ(OLDTAB,30,T[*]);                                         00187150
OTPAR2: CLOSE(OLDTAB); CLOSE(PATCH); CLOSE(SOUR1); CLOSE(PFILE);  00187200
IF HAND1T(A) THEN BEGIN %HERE IF "1S" ERROR DURING "1T" HANDLING 00187300
  %HERE ON "1S" ERROR DURING "1P" OR AFTER "1T" HANDLING          00187400
SOUPAR: CLOSE(OLDTAB); CLOSE(PATCH);                               00187500
        CLOSE(SOUR1); CLOSE(PFILE);                               00187600
        DIALOG(1,WORD); %ERROR FILE NEEDED                       00187700
        IF HAND1S(A,B,F) THEN GO QUIRMDST;                       00187800
END;                                                                00188100
HAND1P(SORTA,MAXCNT); %RESTART "1P" HANDLER                        00188300
END; % "1P" HANDLER                                               00188400
%.....                                                             00188500
%*****PROGRAM STARTS HERE*****                                   00188600
%*****PROGRAM STARTS HERE*****                                   00188700
%.....                                                             00188800
%.....                                                             00188900
BEGIN % INNER BLOCK                                               00189000
EOFMARK:=100000000;                                                00189100
A[0]:=0; A1:=WHEREIS(A[0]); A4:=WHEREIS(A[29]); % SET ADDRESSES    00189200
B[0]:=0; B1:=WHEREIS(B[0]); % "B" ADDRESSES                       00189300
DISKWAIT(1,A,30,COMMON);                                           00189400
COMMON:=0;                                                         00189500
MOVE(26,A[0],P[0]); %SAVE INPUT PARAMETERS FROM CANDE            00189600
USER:=A[6];                                                         00189700
MAXCNT := A[7]+1; %NU OF CHANGES TO WORKFILE                     00189800
LINE:=DECONV(P[5],[40:8]);                                          00189900
MOVE(16,A[10],BASE[0]);                                            00190000
%.....                                                             00190100
%.....                                                             00190300
%*****PROGRAM CONTROL ON INITIAL ERROR*****                     00190400
% PROGRAM CONTROL ON INITIAL ERROR                                00190500
%*****PROGRAM CONTROL ON INITIAL ERROR*****                     00190600
%.....                                                             00190700
IF P[1]=0 THEN                                                      00190800
  IF P[3]="LIST " THEN P[1]:=" "&"1P"[6:36:12]&LINE[18:30:18] 00190850

```

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


```
ELSE P[1]:=P[8];                                00190900
IF P[1],[6:12]="1P" THEN BEGIN                    00191000
IF P[3]="LIST" THEN                              00191100
BEGIN %HERE IF "1P" ERROR INITIALLY             00191200
SRCENAME:=P[8];                                00191300
ONAME:=" "&"1S"[6:36:12]&LINE[18:30:18];        00191400
WRKFILE:=TRUE;                                  00191500
IF P[5],[34:6] NEQ 8 THEN SFLG:=TRUE; %TYPE SEQ 00191600
DIALOG(3,WORD); %RECOVERY FILE NEEDED          00191700
VARNAME:=WORD[1];                               00191800
HAND1P(SORTA,MAXCNT);                           00191900
END                                               00192000
ELSE                                             00192100
GO OUTRMOST;                                    00192300
END;                                             00192500
IF P[1],[6:12]="1T" THEN BEGIN %HERE IF "1T" ERROR INITIALLY 00192600
SRCENAME:=P[8];                                00192700
ONAME:=" "&"1S"[6:36:12]&LINE[18:30:18];        00192800
WRKFILE:=TRUE;                                  00192900
IF HAND1(A) THEN BEGIN %HERE IF "1S" ERROR     00193000
%DURING "1T" PROCESSING                        00193100
DIALOG(1,WORD); %ERROR FILE                    00193200
IF HAND1S(A,B,P) THEN GO OUTRMOST;             00193300
END;                                             00193700
GO OUTR;                                        00193900
END;                                             00194000
%HERE IF "1S" ERROR INITIALLY                   00194100
IF (WRKFILE:=BOCLEAN(P[5],[1:1])) THEN BEGIN %HERE IF WORKFILE 00194200
SRCENAME:=P[1];                                00194300
ONAME:=" "&"1S"[6:36:12]&LINE[18:30:18];        00194400
DIALOG(1,WORD);                                00194500
END                                             00194600
ELSE                                             00194700
BEGIN %NO UPDATE YET                            00194800
SRCENAME:=P[1];                                00194900
DIALOG(2,WORD);                                00195000
ONAME:=WORD[1];                                00195100
END;                                             00195200
IF HAND1S(A,B,P) THEN GO OUTRMOST;             00195300
GO OUTR;                                        00195600
END BLOCK;                                     00195700
OUTRMOST: MESS(A); WRITE(TWX,7,A[*]); %ERROR OUT 00195800
OUTR: S15(A); WRITE(TWX,5,A[*]); %OK OUT       00195900
END PROGRAM.                                   00196100
END;END. LAST CARD ON OCRDING TAPE            99999999
```

LABEL 000000000PRINTER0017510000 EX OBJECT/READ;FILE SOURCEFILE=SYMBOL/HELP;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

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.