

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

OBJECT /READ

SYMBOL/REPLACE

Data Documents, Inc.

```

BEGIN REAL COMMON;
***** REPLACE/CANDE 3-70 *****
COMMENT: * TITLE: B5500/B5700 MARK XIV SYSTEM RELEASE *
* FILE ID: SYMBOL/REPLACE 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 AA332366 AA386657 *;
DEFINE EOR=O&3"37"[1:43:5]#, MAXRANGES = 25#, LEFTARROW="←"#;
ARRAY INBUF[0:10], GRPTKNS,FILINX,REPINX[0:63], FNDINX[0:170];
ARRAY LBOUND,UBOUND[0:MAXRANGES];
SAVE ARRAY ESP,WORK[0:30],OUTBUF[0:111];
BOOLEAN FOUND, BRAAK, PRNT, TEXT, OVERFLO, SEQCHK, SFLG;
FILE IN OLD DISK SERIAL(1,0,0);
FILE OUT NEW DISK SERIAL [20:600](2,10,300,SAVE 1);
INTEGER OLDSIZE, TOKENCOUNT, TOSCAN;
REAL COLS, COLSLEFT, COUNT, ENDWORD, ESPPTR, ERRSW, FLG,
GRPCOUNT, I, LREC, LIMIT, NF, NRANGES, NUM1, NUM2,
LINSIZ, PREVNF, PERLINE, RC, RECSIZ, SCANNED, SEQ, STRTWORD,
TKNS, TOKENLENGTH, TOKENPTR, TOTAL, W1, W2, W3, W4, WORKPTR;
LABEL COPYTOEOF, ENDLOOP, EXIT, FINDLIT, READIN, SAVEEXIT,
TOKENLOOP, WRITENEW;
*****
PROCEDURE TWXOUT(A,N,T); VALUE N,T; REAL A,N,T;
BEGIN COMMUNICATE(-11); BRAAK:=BOOLEAN(T); END;
*****
REAL STREAM PROCEDURE ADDRESS(A);
BEGIN SI:=A; ADDRESS:=SI; END;
*****
REAL STREAM PROCEDURE DECCONV(OCTV); VALUE OCTV;
BEGIN SI:=LOC OCTV; DI:=LOC DECCONV; DS:=8DEC; END;
*****
STREAM PROCEDURE MOVEWORDS(N,A,B); VALUE N;
BEGIN SI:=A; DI:=B; DS:=N WDS; END;
*****
REAL STREAM PROCEDURE OCTCONV(DECV);
BEGIN SI:=DECV; DI:=LOC OCTCONV; DS:= 8 OCT; END;
*****
PROCEDURE DISKWAIT(IC,AREA,WDS,ADDR);
VALUE IO,WDS,ADDR; REAL IO,WDS,ADDR; ARRAY AREA[0]; COMMUNICATE(-8);
*****
STREAM PROCEDURE ERRMSG (BUF, ERRMSG); VALUE ERRMSG;
*****
BEGIN
DI:=BUF; DS:=4LIT"ERR:"; SI:=LOC ERRMSG; SI:=SI+1; DS:=7CHR;
DS:=LIT LEFTARROW;
END ERRMSG;
*****
STREAM PROCEDURE NUMFOUND(O,NM); VALUE NM;
*****
BEGIN
DI:=0; DS:=29LIT"NUMBER OF STRINGS REPLACED = ";
SI:=LOC NM; DS:=8 DEC; DI:=DI-8; DS:=7 FILL;
END STREAM PROCEDURE NUMFOUND;
*****
STREAM PROCEDURE REKNUMBER(BUF,LREC); VALUE LREC;

```

```

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

```

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

```

%*****00015000
BEGIN                                00015100
SI:=LOC LREC; DI:=BUF; DS:=8DEC; DI:=BUF; DS:=7FILL; 00015200
SI:=BUF; DI:=BUF; 8(IF SC=" " THEN SI:=SI+1 ELSE DS:=CHR); 00015300
DS:=8LIT" ";                          00015400
END STREAM PROCEDURE REKNUMBER;        00015500
%*****00015600
STREAM PROCEDURE OUTFORMAT(OUTBUF,INBUF); 00015700
%*****00015800
BEGIN LOCAL SV;                       00015900
SI:=INBUF; 2(SI:=SI+36); DI:=OUTBUF; SV:=DI; DS:=WDS; DI:=SV; 00016000
DS:=7FILL; SI:=SV; DI:=SV; 8(IF SC=" " THEN SI:=SI+1 ELSE DS:=CHR); 00016100
DS:=LIT" "; SI:=INBUF; 2(DS:=36 CHR); DS:=9LIT" "; 00016200
END STREAM PROCEDURE OUTFORMAT;        00016300
%*****00016400
STREAM PROCEDURE TRUNCMSG(OUTBUF,SFLG,LREC,SEQ); VALUE SFLG; 00016500
%*****00016600
BEGIN LOCAL SV; LABEL EXIT;           00016700
DI:=OUTBUF; DS:=8LIT" ** LINE "; SV:=DI; 00016800
SFLG(SI:=SEQ; DS:=8CHR; DI:=SV; DS:=7FILL; SI:=SV; DI:=SV; 00016900
8(IF SC=" " THEN SI:=SI+1 ELSE DS:=CHR); 00017000
JUMP OUT TO EXIT;                      00017100
SI:=LREC; DS:=8DEC; DI:=SV; DS:=7FILL; SI:=SV; DI:=SV; 00017200
8(IF SC=" " THEN SI:=SI+1 ELSE DS:=CHR); 00017300
EXIT:                                   00017400
DS:=19LIT" TRUNCATED. ";               00017500
END STREAM PROCEDURE TRUNCMSG;          00017600
%*****00017700
REAL STREAM PROCEDURE SETNUM(BUF,SEQ,LREC,SFLG,STAR); VALUE SFLG,STAR; 00017800
%*****00017900
BEGIN LOCAL SV; LABEL LBL;            00018000
DI:=BUF; DS:=LIT" "; TALLY:=1;         00018100
STAR(DS:=LIT" *"; TALLY:=2);           00018200
SV:=DI;                                 00018300
SFLG(SI:=SEQ; DI:=LOC STAR; DS:=8CHR; JUMP OUT TO LBL); 00018400
SI:=LREC; DI:=LOC STAR; DS:=8DEC;      00018500
LBL:                                     00018600
DI:=LOC STAR; DS:=7FILL; SI:=LOC STAR; 00018700
DI:=SV;                                 00018800
8(IF SC=" " THEN SI:=SI+1 ELSE         00018900
  BEGIN DS:=CHR; TALLY:=TALLY+1; END); 00019000
SETNUM := TALLY;                        00019100
END STREAM PROCEDURE SETNUM;           00019200
%*****00019300
REAL STREAM PROCEDURE ATOKEN(IPTR,IPTR,EQR); VALUE EUR; 00019400
%*****00019500
BEGIN LOCAL STRT,FIN; LABEL LOOP,NEXTADRS,EXIT; 00019600
SI:=IPTR; SI:=SI+5; SI:=SC;           00019700
LOOP: IF SC=" " THEN BEGIN SI:=SI+1; GO TO LOOP; END; 00019800
STRT:=SI; % SAVE STARTING ADDRESS      00019900
IF SC GEQ "0" THEN IF SC LEQ "9" THEN; 00020000
IF TOGGLE THEN % NUMERIC TOKEN        00020100
  BEGIN                                00020200
  SI:=SI+1; TALLY := 1;                00020300
  62(IF SC GEQ "C" THEN IF SC LEQ "9" THEN; 00020400
  IF TOGGLE THEN BEGIN SI:=SI+1; TALLY:=TALLY+1; END ELSE JUMP OUT); 00020500
  GO TO NEXTADRS;                       00020600
  END; % IF NUMERIC TOKEN               00020700
IF SC = ALPHA THEN IF SC LEQ "9" THEN % ALPHANUMERIC TOKEN 00020800
  BEGIN                                  00020900

```

```

SI:=SI+1; TALLY:=1;                                00021000
62(IF SC = ALPHA THEN IF SC LEQ "9" THEN;          00021100
IF TOGGLE THEN BEGIN SI:=SI+1; TALLY:=TALLY+1; END ELSE JUMP OUT);00021200
GO TO NEXTADRS;                                    00021300
END; % IF ALPHANUMERIC TOKEN                       00021400
IF SC NEQ LEFTARROW THEN % SPECIAL CHARACTER       00021500
BEGIN TALLY:=1; SI:=SI+1; GO TO NEXTADRS; END;    00021600
SI:=LOC STRT; DI:=LOC EOR;                          00021700
IF 8 SC=DC THEN GO TO EXIT; % END OF RECORD       00021800
SI:=STRT; SI:=SI+1; TALLY:=1;                     00021900
NEXTADRS:                                          00022000
FIN:=SI; SI:=LOC FIN; DI:=IPTR; DS:=WDS; % ADDRESS OF NEXT CHR. 00022100
SI:=LOC STRT; DI:=TPTR; DS:=WDS; % ADDRESS OF TOKEN 00022200
EXIT:                                              00022300
ATOKEN := TALLY; % TOKEN SIZE, ( ZERO IF EOR ) '   00022400
END STREAM PROCEDURE ATOKEN;                       00022500
%*****00022600
REAL STREAM PROCEDURE LITSCAN(TS,WP,FINX,F); VALUE TS; 00022700
%*****00022800
BEGIN LOCAL FLOK,OS;                               00022900
DI:=F; DS:=BLIT"0"; % RESET F TO FALSE            00023000
SI:=FINX; SI:=SI+4; DI:=LOC OS; DI:=DI+7; DS:=CHR; % OLD SIZE 00023100
SI:=SC; FLOK:=SI; % ADDRESS OF OLD STRING         00023200
SI:=WP; SI:=SI+5; SI:=SC; % ADDRESS OF WORK AREA 00023300
TS(LITSCAN:=SI; DI:=FLOK;                          00023400
IF DS SC = DC THEN JUMP OUT;                       00023500
SI:=LITSCAN; SI:=SI+1; TALLY:=TALLY+1);          00023600
IF TOGGLE THEN % FOUND A MATCH                    00023700
BEGIN DI:=F; DI:=DI+7; DS:=LIT"1"; SI:= LITSCAN; END; 00023800
LITSCAN:=SI; SI:=LOC LITSCAN; DI:=WP; DS:=WDS;    00023900
LITSCAN:=TALLY;                                    00024000
END STREAM PROCEDURE LITSCAN;                      00024100
%*****00024200
BOOLEAN STREAM PROCEDURE TOKENSMATCH(N,NDX1,NDX2); VALUE N; 00024300
%*****00024400
BEGIN                                               00024500
LABEL EXIT;                                        00024600
LOCAL S,D,R;                                       00024700
SI:=NDX1; SI:=SI+4; DI:=NDX2; DI:=DI+4;          00024800
N(S:=SI; D:=DI;                                    00024900
IF SC=DC THEN % TOKEN LENGTHS ARE EQUAL.         00025000
BEGIN                                              00025100
SI:=SI-1; DI:=LOC R; DI:=DI+7; DS:=CHR;         00025200
SI:=SC; DI:=D; DI:=DI+1; DI:=DC;               00025300
IF R SC NEQ DC THEN JUMP OUT TO EXIT; % TOKENS NOT EQUAL. 00025400
END ELSE JUMP OUT TO EXIT;                       00025500
SI:=S; SI:=SI+8; DI:=D; DI:=DI+8;))            00025600
TALLY:=1;                                         00025700
EXIT: TOKENSMATCH:=TALLY;                         00025800
END OF TOKENSMATCH;                               00025900
%*****00026000
STREAM PROCEDURE REPLACELIT(W1,W2,W3,WORKPTR,OSIZ,REPINX,OVERFLO); 00026100
%*****00026200
VALUE W1,W2,W3,OSIZ;                               00026300
COMMENT W1 IS ADDRESS OF WORK[0],                00026400
W2 IS ADDRESS OF WORK[9] IF SEQUENTIAL FILE, OF WORK[10] IF DATA FILE 00026500
W3 IS ADDRESS OF WORK[20]                        00026600
WORKPTR CONTAINS ADDRESS OF NEXT CHARACTER IN WORK RECORD 00026700
OSIZ IS SIZE ( NO. OF CHARACTERS ) IN OLD STRING 00026800
END OF COMMENT;                                   00026900

```

```

BEGIN LOCAL NSIZ,SV; LABEL LBL;                                00027000
DI:=W2; DS:=8LIT" "; SI:=W2; DS:=10 WDS; % BLANK OUT WORK AREA 00027100
SI:=WORKPTR; SI:=SI+5; SI:=SC; SV:=SI; % LOCATION OF OLD STRING 00027200
SI:=SI+OSIZ; DI:=W3; 2(DS:=40CHR); % SAVE REMAINDER OF RECORD 00027300
SI:=REPINX; SI:=SI+4; DI:=LOC NSIZ; DI:=DI+7; DS:=CHR; % NEW SIZE 00027400
SI:=SC; DI:=SV; DS:=NSIZ CHR; SV:=DI; % REPLACE OLD WITH NEW 00027500
SI:=W3; 2(DS:=40CHR); % REPLACE REMAINDER OF RECORD 00027600
SI:=LOC SV; DI:=WORKPTR; DS:=WDS; % ADRS OF NEXT CHAR. TO BE SCANNED 00027700
SI:=W2; 2(40(IF SC=" " THEN SI:=SI+1 ELSE JUMP OUT 2 TO LBL)); 00027800
LBL;                                                            00027900
IF TOGGLE THEN ELSE % OVERFLOW OCCURRED                       00028000
  BEGIN DI:=OVERFLO; DS:=8LIT"00000001"; END;                   00028100
END STREAM PROCEDURE REPLACELIT;                                00028200
%*****00028300
STREAM PROCEDURE REPLACETOKENS(W1,W2,W3,W4,FILINX,NTKNS,REPINX,OVERFLO);00028400
%*****00028500
VALUE W1,W2,W3,NTKNS;                                          00028600
COMMENT                                                         00028700
W1 IS ADDRESS OF WORK[0],                                       00028800
W2 IS ADDRESS OF WORK[9] IF SEQUENTIAL FILE, OF WORK[10], IF DATA FILE 00028900
W3 IS ADDRESS OF WORK[20],                                       00029000
W4 IS ASSIGNED ADDRESS OF FIRST CHARACTER FOLLOWING REPLACEMENT STRING 00029100
END OF COMMENT;                                                00029200
BEGIN LOCAL OSIZ,NSIZ,SV; LABEL LBL;                            00029300
DI:=W2; DS:=8LIT" "; SI:=W2; DS:=10WDS; % BLANK OUT WORK AREA 00029400
SI:=FILINX; NTKNS(SI:=SI+8); % SKIP TO END OF TOKEN GROUP      00029500
SI:=SI-4; DI:=LOC OSIZ; DI:=DI+7; DS:=CHR; % LENGTH OF LAST TOKEN 00029600
SI:=SC; SI:=SI+OSIZ; % POINT TO 1ST CHAR. FOLLOWING EXIST. STRING 00029700
DI:=W3; 2(DS:=40CHR); % SAVE PORTION OF RECORD FOLLOWING EXIST. STRING 00029800
SI:=REPINX; SI:=SI+4; DI:=LOC NSIZ; DI:=DI+7; DS:=CHR; % REPL. SIZ 00029900
SI:=SC; % POINT TO ACTUAL REPLACEMENT STRING ( IN ESP RECORD ) 00030000
DI:=FILINX; DI:=DI+5; DI:=DC; % POINT TO FIRST CHAR. OF EXIST. STRING 00030100
DS:=NSIZ CHR; % MOVE REPLACEMENT STRING TO WORK RECORD        00030200
SV:=DI; SI:=LOC SV; DI:=W4; DS:=WDS; % ADDRESS AFTER REPLACEMENT 00030300
DI:=SV; SI:=W3; 2(DS:=40CHR); % REPLACE REMAINDER OF RECORD 00030400
SI:=W2; 2(40(IF SC=" " THEN SI:=SI+1 ELSE JUMP OUT 2 TO LBL)); 00030500
LBL;                                                            00030600
IF TOGGLE THEN ELSE % OVERFLOW OCCURRED                       00030700
  BEGIN DI:=OVERFLO; DS:=8LIT"00000001"; END;                   00030800
END STREAM PROCEDURE REPLACETOKENS;                             00030900
%*****00031000
PROCEDURE PRINT;                                               00031100
%*****00031200
BEGIN                                                            00031300
IF NOT BRAAK THEN                                             00031400
  BEGIN                                                         00031500
IF OVERFLO THEN % SEND TRUNCATION MESSAGE                     00031600
  BEGIN                                                         00031700
IF LINSIZ GTR 0 THEN % TERMINATE CURRENT LINE                 00031800
  BEGIN                                                         00031900
    TWXOUT(OUTBUF[0],0,1); LINSIZ:=0;                          00032000
  END;                                                           00032100
  TRUNCMSG(OUTBUF,SFLG,LREC,INBUF[9]);                          00032200
  TWXOUT(OUTBUF[0],27,1);                                        00032300
  END;                                                           00032400
IF PRNT THEN                                                  00032500
  BEGIN                                                         00032600
IF TEXT THEN                                                  00032700
  BEGIN                                                         00032800
IF NOT SFLG THEN % DATA RECORDS                             00032900

```

1	BEGIN	00033000
2	REKNUMBER(OUTBUF,LREC); TWXOUT(OUTBUF[0],8,1);	00033100
3	TWXOUT(INBUF[0],80,1);	00033200
4	END	00033300
5	ELSE	00033400
6	BEGIN	00033500
7	OUTFORMAT(OUTBUF,INBUF);	00033600
8	TWXOUT(OUTBUF[0],81,1);	00033700
9	END;	00033800
10	END % IF TEXT	00033900
11	ELSE	00034000
12	BEGIN	00034100
13	COUNT:=SETNUM(OUTBUF,INBUF[9],LREC,SFLG,(PERLINE GTR 1));	00034200
14	TWXOUT(OUTBUF[0],COUNT,3"100000");	00034300
15	IF LINSIZ := LINSIZ + COUNT GTR 60 THEN	00034400
16	BEGIN	00034500
17	TWXOUT(OUTBUF[0],0,1); LINSIZ := 0;	00034600
18	END;	00034700
19	END; % IF NOT TEXT;	00034800
20	END; % IF PRNT	00034900
21	END; % IF NOT BRAAK;	00035000
22	END PROCEDURE PRINT;	00035100
23	*****	00035200
24	REAL STREAM PROCEDURE TSIZE(WORK);	00035300
25	*****	00035400
26	BEGIN	00035500
27	SI:=WORK; SI:=SI+4; DI:=LOC TSIZE; DI:=DI+4; DS:=4CHR;	00035600
28	END STREAM PROCEDURE TSIZE;	00035700
29	*****	00035800
30	REAL STREAM PROCEDURE UNPAK(ADRS,FINX,NF,RINX,RC,TKN,NUM1,NUM2,FLG);	00035900
31	*****	00036000
32	COMMENT ROUTINE TO TRANSLATE CODED STRINGS IN INPUT SAVE ARRAY	00036100
33	INTO USEABLE FORM FOR "REPLACE" PROGRAM.	00036200
34	FORMAT OF INPUT ARRAY IS:	00036300
35	WD[0],[11:01] = 1 IF "PRINT" SPECIFIED	00036400
36	WD[0],[17:01] = 1 IF "PRINT TEXT" SPECIFIED	00036500
37	WD[0],[24:24] = TOTAL SIZE (IN CHARACTERS) OF INPUT ARRAY (OCTAL)	00036600
38	FORMAT OF INPUT ARRAY, STARTING AT WORD 1:	00036700
39	EACH "STRING" IS PRECEDED BY A FOUR CHARACTER CODE -	00036800
40	CHARACTER 1 = 1 IF AN "EXISTING" STRING	00036900
41	CHARACTER 1 = 2 IF A "REPLACEMENT" STRING	00037000
42	CHARACTER 1 = 3 IF A SINGLE SEQUENCE NUMBER	00037100
43	CHARACTER 1 = 4 IF A LOWER BOUND FOR A SEQUENCE RANGE	00037200
44	CHARACTER 2 = 1 IF "FIRST" IS SPECIFIED	00037300
45	CHARACTER 3 = 1 IF "LITERAL" IS SPECIFIED	00037400
46	CHARACTER 4 = SIZE (IN CHARACTERS) OF THE STRING (OCTAL)	00037500
47	CHARACTER 5 = FIRST CHARACTER OF THE STRING	00037600
48	FORMAT OF OUTPUT ARRAY WORDS:	00037700
49	WD.[17,01] = 1 IF "FIRST" SPECIFIED	00037800
50	WD.[23:01] = 1 IF "LITERAL" SPECIFIED	00037900
51	WD.[24:06] = SIZE OF STRING OR TOKEN	00038000
52	WD.[30:18] = ADDRESS OF STRING OR TOKEN	00038100
53	THE ROUTINE -	00038200
54	BREAKS THE "EXISTING" STRINGS INTO TOKENS (UNLESS "LITERAL" IS	00038300
55	SPECIFIED) AND PLACES THE SIZE AND LOCATION (ABSOLUTE ADDRESS	00038400
56	OF THE TOKEN IN THE INPUT ARRAY) INTO ARRAY "FINX".	00038500
57	THE NUMBER OF TOKENS INTO WHICH THE STRING IS DIVIDED IS	00038600
58	PASSED BACK TO PARAMETER "TKN", AND THE WORD INDEX (PARAMETER	00038700
59	"NF") IS INCREMENTED BY 1 FOR EACH TOKEN PLACED IN "FINX".	00038800
60	THE SIZE AND LOCATION OF THE "REPLACEMENT" STRINGS ARE PLACED	00038900

```

IN ARRAY RINX IN THE SAME MANNER ( EXCEPT THAT THEY ARE NOT          00039000
BRUKEN DOWN INTO TOKENS ) AND THE WORD INDEX ("RC") INCREMENTED      00039100
BY 1 FOR EACH "REPLACEMENT" STRING.                                  00039200
WHEN A SEQUENCE NUMBER IS ENCOUNTERED=                               00039300
FLG = 1 FOR A SINGLE SEQUENCE NUMBER,                               00039400
FLG = 2 FOR A SEQUENCE "RANGE"                                       00039500
NUM2 = OCTAL VALUE OF SEQUENCE NUMBER ( OR UPPER BOUND FOR RANGE )   00039600
NUM1 = OCTAL VALUE OF LOWER BOUND FOR RANGE ( WHERE APPLICABLE ).    00039700
THE POSITION OF THE INPUT STRING TO BE EXAMINED IS SPECIFIED          00039800
BY THE ADDRESS IN PARAMETER "ADRS". THIS PARAMETER IS                00039900
RE=ASSIGNED THE ADDRESS OF THE NEXT INPUT STRING WHEN                00040000
THE PROCEDURE IS EXITED.                                             00040100
THE SIZE OF THE CURRENT STRING IS RETURNED TO "UNPAK"                00040200
TO ALLOW COMPARISON BETWEEN THE TOTAL SIZE OF THE INPUT ARRAY        00040300
AND THE SIZE OF THE STRINGS ALREADY PROCESSED.                       00040400
END OF COMMENT;                                                      00040500
  BEGIN LOCAL N,T1,T2,SV1,SV2,SV3,DV,TK;                               00040600
  LABEL CYCLE,XFER,NUMBR,EXIT;                                         00040700
  DI:=FLG; DS:=8LIT"0"; % RESET THE SEQUENCE NUMBER FLAG              00040800
  SI:=ADRS; SI:=SI+5; SI:=SC; SV1:=SI; % STARTING ADDRESS              00040900
  SI:=SI+3; DI:=LOC T1; DI:=DI+7; DS:=CHR; % SIZE OF THIS STRING      00041000
  TALLY:=T1; UNPAK:=TALLY; % SIZE TO "UNPAK" ALSO                    00041100
  SI:=SV1; IF SC GTR "2" THEN GO TO NUMBR; % FIELD IS A SEQUENCE NO.  00041200
  IF SC="1" THEN % AN "EXISTING" STRING                                00041300
    BEGIN SI:=NF; DI:=LOC N; DS:=WDS; % "OLD" STRING INDEX WORD       00041400
    DI:=FINX; N(DI:=DI+8); DV:=DI; % SKIP OVER PREVIOUS WDS, SAVE ADDR 00041500
    SI:=SV1; SI:=SI+2; IF SC="1" THEN % "LITERAL" CODE                 00041600
      BEGIN SI:=SI-1; DI:=DI+2; DS:=3 CHR; % "FIRST","LITERAL" AND SIZE 00041700
      SV2:=SI; SI:=LOC SV2; SI:=SI+5; DS:=3 CHR; % ADDRESS OF STRING   00041800
      TALLY:=N; TALLY:=TALLY+1; N:=TALLY; % INCREMENT INDEX           00041900
      SI:=LOC N; DI:=NF; DS:=WDS; % TRANSFER TO EXTERNAL VARIABLE     00042000
      SI:=SV2; SI:=SI+T1; SV3:=SI; % ADDRESS OF NEXT FIELD            00042100
      TALLY:=1; TK:=TALLY; % "LITERAL" STRING = 1 TOKEN               00042200
      GO TO EXIT;                                                       00042300
    END; % IF "LITERAL"                                                00042400
    % BREAK UP INTO "TOKENS" IF NOT "LITERAL"                           00042500
    SI:=SI+2; SV2:=SI; % POINT TO ACTUAL STRING                         00042600
  CYCLE:                                                                00042700
    SI:=SV2; TALLY:=T1; % SKIP OVER BLANKS NEXT                        00042800
    T1(IF SC NEQ " " THEN JUMP OUT; SI:=SI+1; TALLY:=TALLY+63);        00042900
    T1:=TALLY; SV2:=SI; SV3:=SI;                                       00043000
    SI:=LOC T1; SI:=SI+7; IF SC="0" THEN GO TO EXIT; % END STRING     00043100
    TALLY:=TK; TALLY:=TALLY+1; TK:=TALLY; % TOKEN COUNT, THIS STRING  00043200
    SI:=SV2; IF SC=ALPHA THEN IF SC LSS "0" THEN % STARTS WITH A LETTER 00043300
      BEGIN TALLY:=0;                                                  00043400
      T1(IF SC LEQ "9" THEN ELSE JUMP OUT;                               00043500
      IF SC = ALPHA THEN ELSE JUMP OUT;                                 00043600
      SI:=SI+1; TALLY:=TALLY+1);                                        00043700
      T2:=TALLY; TALLY:=0; SV3:=SI;                                     00043800
      IF TOGGLE THEN T1:=TALLY ELSE % ALL CHARACTERS USED IF TOGGLE   00043900
      BEGIN TALLY:=T1; T2(TALLY:=TALLY+63); T1:=TALLY; END;           00044000
    XFER:                                                                00044100
      SI:=SV1; SI:=SI+1; DI:=DV; DI:=DI+2;                             00044200
      DS:=2CHR; % "FIRST" AND "LITERAL" CODES                          00044300
      SI:=LOC T2; SI:=SI+7; DS:=CHR; % TOKEN SIZE                      00044400
      SI:=LOC SV2; SI:=SI+5; DS:=3 CHR; % ADDRESS OF TOKEN            00044500
      SI:=SV3; SV2:=SI; DV:=DI; % SAVE ADDRESSES                       00044600
      TALLY:=N; TALLY:=TALLY+1; N:=TALLY; % INCREMENT INDEX           00044700
      SI:=LOC N; DI:=NF; DS:=WDS; % TRANSFER TO EXTERNAL VARIABLE     00044800
      GO TO CYCLE;                                                      00044900
    END % IF A LETTER STRING

```

```

ELSE IF SC LEQ "9" THEN * A DIGIT STRING          00045000
  BEGIN TALLY:=0;                                00045100
  T1(IF SC LSS "0" THEN JUMP OUT;                00045200
  IF SC GTR "9" THEN JUMP OUT;                  00045300
  SI:=SI+1; TALLY:=TALLY+1);                    00045400
  T2:=TALLY; TALLY:=0; SV3:=SI;                 00045500
  IF TOGGLE THEN * ALL CHARACTERS NOT USED      00045600
  BEGIN TALLY:=T1; T2(TALLY:=TALLY+63); T1:=TALLY; END 00045700
  ELSE T1:=TALLY;                                00045800
  GO TO XFER; * TRANSFER TOKENS                  00045900
  END IF A DIGIT STRING;                         00046000
  * A "SPECIAL" CHARACTER                        00046100
  TALLY:=T1; TALLY:=TALLY+63; T1:=TALLY; * ONE CHARACTER TOKEN 00046200
  TALLY:=1; T2:=TALLY; SI:=SI+1; SV3:=SI;       00046300
  GO TO XFER;                                    00046400
  END * IF AN "EXISTING" STRING                  00046500
ELSE * A "REPLACEMENT" STRING                   00046600
  BEGIN SI:=RC; DI:=LOC N; DS:=WDS; * "NEW" STRING INDEX WORD 00046700
  DI:=RINX; N(DI:=DI+8); * SKIP OVER PREVIOUS WORDS          00046800
  SI:=SV1; SI:=SI+3; DI:=DI+4; DS:=CHR; * STRING SIZE        00046900
  SV2:=SI; SI:=LOC SV2; SI:=SI+5; DS:=3 CHR; * ADDRESS OF STRING 00047000
  TALLY:=N; TALLY:=TALLY+1; N:=TALLY; * INCREMENT INDEX      00047100
  SI:=LOC N; DI:=RC; DS:=WDS; * TRANSFER TO EXTERNAL VARIABLE 00047200
  SI:=SV2; SI:=SI+T1; SV3:=SI; * ADDRESS OF NEXT FIELD       00047300
  GO TO EXIT;                                        00047400
  END; * IF A "REPLACEMENT" STRING                00047500
NUMBR; * A SEQUENCE NUMBER                        00047600
  DI:=FLG; DS:=BLIT"00000001"; * SEQUENCE NUMBER FLAG        00047700
  IF SC="4" THEN * SEQUENCE "RANGE" SPECIFIED              00047800
  BEGIN DI:=FLG; DI:=DI+7; DS:=LIT"2"; * CODE FOR "RANGE"    00047900
  SI:=SI+3; DI:=LOC N; DI:=DI+7; DS:=CHR; * SIZE OF FIELD    00048000
  DI:=NUM1; DS:=N OCT;                                    00048100
  SI:=SI+3; DI:=LOC N; DI:=DI+7; DS:=CHR; * SIZE OF NEXT FIELD 00048200
  TALLY:=UNPAK; TALLY:=TALLY+4; TALLY:=TALLY+N; UNPAK:=TALLY; 00048300
  DI:=NUM2; DS:=N OCT; SV3:=SI; GO TO EXIT;                00048400
  END; * IF "RANGE" SPECIFIED                          00048500
  SI:=SI+3; DI:=LOC N; DI:=DI+7; DS:=CHR; * SIZE OF FIELD    00048600
  DI:=NUM2; DS:=N OCT; SV3:=SI;                        00048700
EXIT; SI:=LOC SV3; DI:=ADRS; DS:=WDS; * ADDRESS OF NEXT FIELD 00048800
  SI:=LOC TK; DI:=TKN; DS:=WDS; * TRANSFER TOKEN COUNT       00048900
  END STREAM PROCEDURE UNPAK;                          00049000
  *****00049100
  ESP[0]:=0; DISKWAIT(1,ESP,30,COMMON);                00049200
  COMMON := ESP[0];                                    00049300
  FILL OLD WITH ESP[5],ESP[6];                          00049400
  FILL NEW WITH ESP[3],ESP[4];                           00049500
  SFLG := ESP[7] NEQ 0; * NOT TYPE DATA FILE             00049600
  RECSIZ := 10 - REAL(SFLG);                             00049700
  DISKWAIT(1,ESP,30,COMMON); COMMON:=0;                00049800
  TOTAL := TSIZE(ESP); * TOTAL NO. OF CHARACTERS IN INPUT RECORD 00049900
  ESPPTR := ADDRESS(ESP[1]); * STARTING ADDRESS OF STRINGS   00050000
  GRPCOUNT := RC := NF := PREVNF := NRANGES := 0; * INITIALIZE 00050100
  WHILE TOTAL GTR 0 DO * DECODE THE INPUT RECORD           00050200
  BEGIN TOTAL := TOTAL - 4 - * DECREMENT TOTAL             00050300
  UNPAK(ESPPTR,FNDINX,NF,REPINX,RC,TKNS,NUM1,NUM2,FLG); 00050400
  IF NF GTR PREVNF THEN * AN "EXISTING" STRING FOUND      00050500
  BEGIN IF NF GTR 100 THEN * TOO MANY TOKENS              00050600
  BEGIN ERRMSG(OUTBUF,"TOOLONG");                        00050700
  TWXOUT(OUTBUF[0],13,0); COMMON.[2:1]:=1; GO TO EXIT;   00050800
  END IF TOO MANY TOKENS;                                00050900

```

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

1	GRPTKNS[GRPCOUNT := GRPCOUNT + 1] := TKNS; % TOKENS/GROUP	00051000	1
2	PREVNF := NF;	00051100	2
3	END IF AN EXISTING STRING;	00051200	3
4	IF FLG GTR 0 THEN % SEQUENCE NUMBER(S) IN INPUT RECORD	00051300	4
5	BEGIN NRANGES := NRANGES + 1;	00051400	5
6	IF NRANGES GTR MAXRANGES THEN % TOO MANY SEQUENCE NUMBERS	00051500	6
7	BEGIN ERRMSG(OUTBUF,"TOOMANY");	00051600	7
8	TWXOUT(OUTBUF[0],13,0); COMMON.[2:1]:=1; GO TO EXIT;	00051700	8
9	END IF TOO MANY SEQUENCE NUMBERS;	00051800	9
10	LBOUND[NRANGES]:=UBOUND[NRANGES]:=NUM2;	00051900	10
11	IF FLG GTR 1 THEN LBOUND[NRANGES]:=NUM1; % SEQ. RANGE	00052000	11
12	END; % IF SEQUENCE NUMBER(S).	00052100	12
13	IF RC GTR 62 THEN % TOO MANY "REPLACEMENT" STRINGS	00052200	13
14	BEGIN ERRMSG(OUTBUF,"TOOLONG");	00052300	14
15	TWXOUT(OUTBUF[0],13,0); COMMON.[2:1]:=1; GO TO EXIT;	00052400	15
16	END IF TOO MANY REPLACEMENT STRINGS;	00052500	16
17	END OF WHILE STATEMENT;	00052600	17
18	LREC := WORK[0] := 0;	00052700	18
19	W1 := ADDRESS(WORK[0]);	00052800	19
20	W2 := ADDRESS(WORK[10-REAL(SFLG)]);	00052900	20
21	W3 := ADDRESS(WORK[20]);	00053000	21
22	PRNT := BOOLEAN(ESP[0].[11:1]); % PRINT SPECIFICATION	00053100	22
23	TEXT := BOOLEAN(ESP[0].[17:1]); % TEXT OR SEQ. NUMBERS	00053200	23
24	RC:=1; % STARTING RANGE	00053300	24
25	SEQCHK := NRANGES GTR 0;	00053400	25
26	READIN;	00053500	26
27	READ(OLD,10,INBUF[*])[SAVEEXIT]; % GET NEXT RECORD	00053600	27
28	LREC := LREC + 1;	00053700	28
29	OVERFLO := FALSE;	00053800	29
30	PERLINE := 0;	00053900	30
31	IF SEQCHK THEN % CHECK SEQUENCE NUMBERS	00054000	31
32	BEGIN	00054100	32
33	SEQ := IF SFLG THEN OCTCONV(INBUF[9]) ELSE LREC;	00054200	33
34	WHILE SEQ GTR UBOUND[RC] DO	00054300	34
35	BEGIN RC := RC + 1;	00054400	35
36	IF RC GTR NRANGES THEN GO COPYTOEOF;	00054500	36
37	END WHILE STATEMENT;	00054600	37
38	IF SEQ LSS LBOUND[RC] THEN GO TO WRITENEW; % SKIP THIS RECORD	00054700	38
39	END; % IF SEQUENCE CHECK	00054800	39
40	MOVEWORDS(RECSIZ,INBUF,WORK); WORK[RECSIZ] := EOR; % XFER TO WORK	00054900	40
41	ENDWORD := -1;	00055000	41
42	FOR I:=1 STEP 1 UNTIL GRPCOUNT DO % CHECK ALL SUBSTITUTIONS	00055100	42
43	BEGIN	00055200	43
44	WORKPTR := W1; % START OF RECORD	00055300	44
45	STRTWORD := ENDWORD + 1; % LOCATION IN ENDINX FOR GROUP	00055400	45
46	ENDWORD := ENDWORD + GRPTKNS[I];	00055500	46
47	TOKENCOUNT := COLS := 0; % RESET	00055600	47
48	IF BOOLEAN(FNDINX[STRTWORD].[23:1]) THEN GO FINDLIT; % LITERAL	00055700	48
49	TOKENLOOP:	00055800	49
50	IF TOKENLENGTH := ATOKEN(WORKPTR,TOKENPTR,W2) = 0 THEN	00055900	50
51	GO TO ENDILOOP; % END OF RECORD	00056000	51
52	FILINX[TOKENCOUNT]:=TOKENPTR & TOKENLENGTH[24:42:6]; % LOCATION	00056100	52
53	TOKENCOUNT := TOKENCOUNT + 1;	00056200	53
54	IF TOKENCOUNT LSS GRPTKNS[I] THEN GO TO TOKENLOOP;	00056300	54
55	IF TOKENSMATCH(TOKENCOUNT,FNDINX[STRTWORD],FILINX) THEN	00056400	55
56	BEGIN % FOUND A MATCH	00056500	56
57	REPLACETOKENS(W1,W2,W3,W4,FILINX,TOKENCOUNT,REFINX[I-1],OVERFLO);	00056600	57
	WORK[RECSIZ] := EOR; WORKPTR := W4;	00056700	
	COMMON := COMMON + 1; % NUMBER OF SUBSTITUTIONS	00056800	
	PERLINE := PERLINE + 1; % NUMBER IN THIS RECORD	00056900	

1	TOKENCOUNT := 0; % RESET	00057000	1
2	IF BOOLEAN(FNDINX[STRTWORD].[17:1]) THEN GO ENDILOOP; % FIRST	00057100	2
3	IF OVERFLO OR W4.[33:15] GTR W3 THEN	00057200	3
4	BEGIN MOVEWORDS(RECSIZ,WORK,INBUF); PRINT; GO WRITENEW; END;	00057300	4
5	GO TO TOKENLOOP; % LOOK AT REMAINDER OF RECORD	00057400	5
6	END IF TOKENS MATCH;	00057500	6
7	TOKENCOUNT := TOKENCOUNT - 1;	00057600	7
8	IF TOKENCOUNT GTR 0 THEN % SHIFT LEFT	00057700	8
9	MOVEWORDS(TOKENCOUNT,FILINX[1],FILINX[0]);	00057800	9
10	GO TO TOKENLOOP; % CONTINUE TO END OF RECORD	00057900	10
11	FINDLIT:	00058000	11
12	COLSLEFT := RECSIZ * 8 - COLS; % CHARACTERS LEFT TO SCAN	00058100	12
13	OLDSIZE := FNDINX[STRTWORD].[24:6]; % LITERAL STRING SIZE	00058200	13
14	IF COLSLEFT LSS OLDSIZE THEN GO TO ENDILOOP; % TOO SMALL	00058300	14
15	TOSCAN := COLSLEFT - OLDSIZE + 1;	00058400	15
16	IF TOSCAN GTR 63 THEN TOSCAN := 63; % 63 IN ONE BREATH	00058500	16
17	SCANNED := LITSCAN(TOSCAN,WORKPTR,FNDINX[STRTWORD],FOUND);	00058600	17
18	COLS := COLS + SCANNED; % UP TO STRING, IF FOUND	00058700	18
19	IF FOUND THEN	00058800	19
20	BEGIN	00058900	20
21	COMMON := COMMON + 1; % TOTAL NO. OF SUBSTITUTIONS	00059000	21
22	PERLINE := PERLINE + 1; % NO. FOR THIS LINE	00059100	22
23	REPLACELIT(W1,W2,W3,WORKPTR,OLDSIZE,REPINX[I-1],OVERFLO);	00059200	23
24	IF OVERFLO OR WORKPTR.[33:15] GTR W2 THEN % TRUNCATION	00059300	24
25	BEGIN MOVEWORDS(RECSIZ,WORK,INBUF); PRINT; GO WRITENEW; END;	00059400	25
26	COLS := COLS + REPINX[I-1].[24:6]; % ADD SIZE OF NEW STRING	00059500	26
27	WORK[RECSIZ]:= EOR; % LEFT ARROW	00059600	27
28	IF PERLINE GTR 71 THEN GO TO ENDILOOP;	00059700	28
29	IF BOOLEAN(FNDINX[STRTWORD].[17:1]) THEN GO TO ENDILOOP; %FRST	00059800	29
30	END IF FOUND;	00059900	30
31	GO TO FINDLIT;	00060000	31
32	END I LOOP;	00060100	32
33	IF PERLINE GTR 0 THEN % RECORD HAS BEEN ALTERED	00060200	33
34	BEGIN MOVEWORDS(RECSIZ,WORK,INBUF);	00060300	34
35	IF PRNT THEN PRINT;	00060400	35
36	END IF ALTERED;	00060500	36
37	WRITENEW:	00060600	37
38	WRITE(NEW,10,INBUF[*]);	00060700	38
39	GO TO READIN;	00060800	39
40	COPYTDEOF:	00060900	40
41	DO BEGIN	00061000	41
42	WRITE(NEW,10,INBUF[*]);	00061100	42
43	READ(OLD,10,INBUF[*])[SAVEEXIT];	00061200	43
44	END UNTIL FALSE;	00061300	44
45	SAVEEXIT:	00061400	45
46	CLOSE(OLD);	00061500	46
47	TWXOUT(OUTBUF[0],0,1);	00061600	47
48	IF COMMON NEQ 0 THEN LOCK(NEW,*);	00061700	48
49	NUMFOUND(OUTBUF,COMMON); TWXOUT(OUTBUF[0],37,1);	00061800	49
50	EXIT;	00061900	50
51	END OF PROGRAM.	00062000	51
52	END;END. LAST CARD ON OCRDING TAPE	00062100	52
53	GO TO INITIATE;	99999999	53
54		19289400	54
55			55
56			56
57			57

LABEL 000000000PRINTER00175100CC EX OBJECT/READ;FILE SOURCEFILE=SYMBOL/REPLACE;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.