

?EXECUTE 0/R

PACKET 25
INPUT 6 CARDS FROM CH
TIME 1128
DATE 78104 FRIDAY, 04/14/78

*** BURROUGHS B5700 DCMCP MARK XVI.0.178 AND INTRINSICS MARK XVI.8.132 ***

#NO MESSAGES TODAY

11:29:00 ?EXECUTE 0/R
11:29:01 ?COMMON= 1
11:29:02 ?FILE LINE= LINE PRINT
11:29:02 ?FILE TTY= TTY DUMMY
11:29:02 ?FILE S=XALGOL/PATCHES
11:29:03 ?END
11:29:04 4:0/R= 2 BOJ 1129 11/22/76
11:29:09 PBD0026 OUT 011 LINE:0/R= 2
11:29:09 DKA IN SER XALGOL PATCHES:0/R= 2
11:30:54 DKA REL XALGOL PATCHES:0/R= 2
11:30:54 PBD0026 REL 011 LINE 1584:0/R= 2
11:30:55 0/R= 2 EOJ 1130
11:30:55 FOR 0/R= 2: PROCESS= 23 SECS, IO= 66 SECS, OLAY= 0
11:30:56 PKT#0025 REMOVED

FILE: XALGOL/PATCHES FRIDAY 04/14/78 11:29 AM

* PATCH 400 FOR XALGOL.XIII CONTAINS 65 CARDS 09417120

COMMENT * * * * * 00490010

* * * * * 00490020

DESCRIPTION OF NCSL PATCHES 00490030

5 / 5 / 72 00490040

* * * * * 00490050

THESE PATCHES ARE INCLUDED TO PROVIDE IMPROVEMENTS 00490060

IN BOTH EFFICIENCY AND USER CONVENIENCE, THEY HAVE BEEN 00490070

COLLECTED FROM A VARIETY OF SOURCES, PRINCIPALLY FROM THE 00490080

WESTINGHOUSE RESEARCH LABORATORY, 00490090

----- 00490100

401 NSRDL-PC 00490110

THIS PATCH REDUCES THE SIZE OF THE DISK AREAS CREATED BY THE 00490120

COMPILER AND SETS THE SAVE TIME OF THE NEWTAPE FILE TO 00490130

15 DAYS AND THE LINE FILE TO 1 DAY, 00490140

----- 00490150

402 NSRDL-PC 1/4/71 DHB 00490160

IN ORDER THAT THE LINE PRINTER FILE WILL NOT BE LOST IF IT 00490170

IS LABELQUATED TO DISK AND THE COMPILER IS DS-ED THIS 00490180

PATCH CAUSES SUCH A LINE FILE TO BE LOCKED AND ENTERED IN 00490190

THE DISK DIRECTORY WHEN THE COMPILER HEADING IS WRITTEN 00490200

BY THE DATIME PROCEDURE, 00490210

----- 00490220

403 NSRDL-PC 1971 00490230

THIS PATCH MODIFIES THE COMPILER LISTING AS FOLLOWS: 00490240

A. A SINGLE SPACED LISTING IS ASSUMED BY DEFAULT AND 00490250

THE \$ OPTION "SINGLE" MUST BE RESET TO GIVE 00490260

DOUBLE SPACING, 00490270

B. THIS ADDS THE ACTUAL PROCESSING TIME USED BY THE COMPILER 00490280

TO THE SUMMARY AND REMOVES THE ESTIMATED AUXILLARY 00490290

MEMORY REQUIRED, 00490300

C. THIS SETS THE COMPILERS ERROR LIMIT VALUE TO A DEFAULT 00490310

VALUE OF 25. THIS VALUE MAY BE CHANGED AT COMPILE TIME 00490320

BY THE "LIMIT" \$ OPTION, 00490330

----- 00490340

404 WR 00490350

THIS PATCH MODIFIES AND CORRECTS THE COMPILERS METHOD OF 00490360

CALCULATING A PROGRAMS CORE ESTIMATE, 00490370

----- 00490380

405 WR: NSRDL-PC DAVE BROWN 00490390

THIS PATCH IMPROVES THE COMPILERS EFFICIENCY BY 00490400

PROVIDING ADDITIONAL SEGMENTATION AND BY REWRITING 00490410

SOME #00R CODING, 00490420

----- 00490430

406 DAVE BROWN 5/5/72 00490440

THIS ALLOWS 8 CHARACTER BCL, 16 OCTADE OCTAL AND 12 HEXADE HEX 00490450

STRINGS SO LONG AS THE FLAG BIT IS NOT SET, 00490460

----- 00490470

407 WR 00490480

THIS IMPLEMENTS THE STORE-STAR OPERATOR OF THE FORM: 00490490

A[I:J] := * + AEXP OR 00490500

B[J] := (*) & AEXP [CONCAT], ETC. 00490510

----- 00490520

Moore Business Forms, Inc. 5/71

```

THE STAR (*) AS THE FIRST "OPERAAND" AFTER THE
REPLACEMENT OPERATOR CAUSES A POLISH(DUP,LOD) TO BE
EMITTED INSTEAD OF RE-EVALUATING THE ENTIRE SUBSCRIPT
EXPRESSION,
-----
408 NSRDL=PC 7/25/71 DMB
TWO NEW FOR=LIST TYPES ARE IMPLEMENTED BY THIS PATCH,
THE FOLLOWING TABLE GIVES EXAMPLES OF THE NEW FOR=LIST
TYPES AND THEIR EQUIVALENT ALGOL-60 CONSTRUCTS,
NEW FOR=LIST TYPE      ALGOL-60 EQUIVALENT
      A(B)C              A STEP B UNTIL C
      A;C                 A STEP 1 UNTIL C
***** END OF NCSL PATCH DESCRIPTIONS *****
S# PATCH 401 FOR XALGOL.XIII CONTAINS 3 CARDS
DEFINE CHUNK = 20#;
FILE OUT LINE DISK SERIAL [20;300] (RR3,15,RR4,SAVE 1);
SAVE FILE OUT NEWTAPE DISK SERIAL [20;600] (RR5,RR6,RR7,SAVE 15);
S# PATCH 402 FOR XALGOL.XIII CONTAINS 6 CARDS
IF Q1=LINE,TYPE=12 OR Q=10 OR Q=13 THEN % GOING TO DISK
BEGIN
  N1:=1; % DONT NAME FILES FOR REMOTES
  LINE,TYPE:=12; WRITE(LINE); LOCK(LINE); % SO ITS NOT LOST ON DS
  LINE,AREAS:=0; LINE,AREASIZE:=0;
END
S# PATCH NUMBER 403 FOR XALGOL CONTAINS 91 CARDS
S# MAXSEG AND IMPROVED SUMMARY ADDED 2/13/74. DMB.
INTEGER MAXSEG; % FOR LARGEST SEGMENT FOUND.
ARRAY LIN[0:9];
STREAM PROCEDURE CODENAME(N1,N2,LIN); VALUE N1,N2;
BEGIN % C O D E / N A M E
  LOCAL N;
  S1:=LOC N1; S1:=S1+1;
  7(IF SC=" " THEN JUMP OUT; TALLY:=TALLY+1; S1:=S1+1);
  N:=TALLY; % NUMBER OF CHARS IN FIRST NAME
  D1:=LIN; DS1=6LIT" "; S1:=LIN; DS1=8 WDS1;
  DI:=LIN; DI:=DI+52; % TO THE "/"
  DI:=DI-N; DI:=DI-N;
  S1:=LOC N1; S1:=S1+1;
  N(DS1=CHR; DI:=DI+1);
  DS1=LIT"/"; DI:=DI+1;
  S1:=LOC N2; S1:=S1+1;
  7(DS1=CHR; DI:=DI+1);
END;
STREAM PROCEDURE UNDERLINE(LIN);
BEGIN
  S1:=LIN; S1:=S1+38;
  DI:=LIN; DI:=DI+38;
  15(IF SC=" " THEN DI:=DI+1 ELSE DS1=LIT" ";
  S1:=S1+2; DI:=DI+1);
END;
STREAM PROCEDURE NAMEIT(A,B,W,LIN); VALUE A,B,W;
BEGIN
  LABEL ZER,ONE,NAM,FIL; LOCAL N;
  DI:=LIN; DI:=DI+37; S1:=LOC A; S1:=S1+1;
  CI:=CI+W; GO TO ZER; GO TO ONE; % GO TO TWO;
  DS1=12LIT"NEW SYMBOLIC"; GO TO NAM;
  ONE; DS1=6LIT"PATCH "; GO TO FIL;

```

```

00490530
00490540
00490550
00490560
00490570
00490580
00490590
00490600
00490610
00490620
00490630
00490640
00490650
01556100
01559000
01560000
01828800
01828810
01828820
01828830
01828840
01828850
00800000
00800010
01693100
01822500
01827010
01827020
01827030
01827040
01827050
01827060
01827070
01827080
01827090
01827100
01827110
01827120
01827130
01827140
01827150
01827160
01827170
01827180
01827190
01827200
01827210
01827220
01827230
01827240
01827250
01827260
01827270
01827280
01827290

```

```

ZER: DSI= 6LIT"SOURCE";
FIL: DSI= 6LIT" FILE ";
NAM: DSI= 3LIT" IS"; DS:=8LIT" ";
7(IF SC=" " THEN JUMP OUT; TALLY:=TALLY+1; SII:=SI+1);
NI=TALLY; SII=SI-NI; DI:=DI-NI; DSI=N CHR;
DSI=LIT"/"; SII=LOC B; SII=SI+1; DSI=7 CHR;
END;
ELSE
BEGIN % GIVE LINE FILE NAME OF CODE FILE
IF STATUS(LIN[*]) > 0 THEN LINE,TYPE:=15; % REMOTE INITIATION
LINE,MFID:=NI; LINE,FID:=N2;
END;
WRITE(LINE[DBL], %
" ",A6,"DAY", "0," "12,"I",A2,X1,A3>,
$ VOID
IF H ≥ 12 THEN "PM," ELSE "AM,");
$ VOIDT 01835999
IF N1 ≥ 0 THEN % NOT FOR REMOTE SO NAME HIS FILES
BEGIN
WRITE(LINE[DBL]);
CODENAME(N1,N2,LIN); WRITE(LINE,9,LIN[*]);
UNDERLINE(LIN); WRITE(LINE[DBL],9,LIN[*]);
NI:=CARD,MFID; N2:=CARD,FID;
IF MERGETOG THEN % $ TAPE
BEGIN
NAMEIT(TAPE,MFID,TAPE,FID,0,LIN);
WRITE(LINE[DBL],9,LIN[*]);
END;
IF CARD,TYPE=12 THEN
NAMEIT(N1,N2,REAL(MERGETOG),LIN)
ELSE
NAMEIT("CARD", "DECK",REAL(MERGETOG),LIN);
WRITE(LINE[DBL],9,LIN[*]);
IF NEWTOG THEN % $ NEWTAPE
BEGIN
NAMEIT(NEWTAPE,MFID,NEWTAPE,FID,2,LIN);
WRITE(LINE[DBL],9,LIN[*]);
END;
WRITE(LINE);
END;
OPTIONWORD:=FALSE&TRUE(SINGLBIT:1); % SET SINGLE RESET REST
OPTION$[43]=1; % FOR SINGLE SPACE TO BE DEFAULT
"6SINGL",1, % 42, 43
SINGLTOG:=TRUE; % DEFAULT SINGLE SPACE LISTING
ERRMAX:=25; % DEFAULT VALUE. USER CAN CHANGE BY $ CARD
FORMAT PAN ( % COMPILER SUMMARY
"NUMBER OF ERRORS DETECTED ="I4", COMPILE TIME ="I5" IN "I6" SECONDS,"
X16,2A4,/,
"PROGRAM SIZE: CARDS PRT DISK SEGS PGM SEGS MAX SEG "
"WD$ OF CODE"/
X14,I5,I7,I10,I11,I10,I12//
"ESTIMATED CORE STORAGE REQUIRED ="I6," WORDS.");
WRITE(LINE,PAN, % COMPILER SUMMARY
ERRORCOUNT,TIME(2)/60,(TIME(1)-TIME1)/60,GT1,GT2,
CARDCOUNT,PRTIMAX,
IF DASCHUNK THEN DA ELSE =CHUNK*ENTIER(=DA/CHUNK),
SGAVL=1,MAXSEG,AKKUM,

```

```

01827300
01827310
01827320
01827330
01827340
01827350
01827360
01828860
01828870
01828880
01828890
01828900
01829000
01832000
01832500
01835000
01835001
01835025
01835050
01835075
01835100
01835125
01835150
01835175
01835200
01835225
01835250
01835275
01835300
01835325
01835350
01835375
01835400
01835425
01835450
01835475
01835500
01835525
01835550
01835575
02130000
02131100
09027044
09028020
09828910
09274400
09774410
09774420
09274430
09774440
09774450
09774460
09417000
09417500
09418000
09418500
09419000

```

Moore Business Forms, Inc. 3V 1412

```

GT11))
END;
IF SIZE>MAXSEG THEN MAXSEG:=SIZE; & LARGEST SO FAR
$# PATCH 404 FOR XALGOL,XIII CONTAINS 7 CARDS
$ VOIDT
GT11:=IF NOOFARRAYS > 10 THEN 5000 ELSE NOOFARRAYS*500;
$ VOID 09361061
GT11:=GT11+(IF AKKUM > 10000 THEN 5000 ELSE AKKUM/2); %
IF (IOTEMP:=GET(L-2)).[46:2] = 0 THEN CURRENT:=IOTEMP
IF (IOTEMP:=GET(L-1)).[46:2] = 0 THEN IOTEMP:=IOTEMP DIV 413148200
$# PATCH 405 FOR XALGOL,XIII CONTAINS 269 CARDS
DEFINE EFFSEGMENT =DEFINE ZOTSEGHHERE=##; % TO FORCE A SEGMENT
$ VOID 01339051
DEFINE FSAVE=RR10#; COMMENT SAVES FRACTIONAL PART EXPONENT ;
INTEGER IDLOC; % POINTER INTO IDARRAY (FILE PARAMETER BLOCK)
$ VOID 01419001
$ VOID 01525001
INTEGER KOUNT;
$ VOID 01601001
$ VOID 01687001
$ VOIDT 01737251
$ VOID 01786001
DEFINE LOOKDEC=% USED ONLY IN SEARCHLIB, SEE 02013184
%; % END OF LOOKDEC DEFINE
LOOKDEC % DECLARS STREAM PROCEDURE LOOK SEE 2001010
$VOID 02023000
$ VOID 02263001
; LOCK(POP[*]); LOCK(WOP[*]);
COUNT:=0; T:=63; %
$ VOIDT
$ VOIDT
$ VOID 03018001
$ VOID 03033001
DEFINE CONSTANTCLEAN=BEGIN IF MRCLEAN THEN CONSTANTCLEANP END#;
PROCEDURE CONSTANTCLEANP; FORWARD;
DEFINE PASSTYPE(E)=((E,CLASS=BOOPROCID).[46:2]);#;
DEFINE SORTSMT= SORTMERGESTMT(FALSE)#;
MERGESTMT= SORTMERGESTMT(TRUE)#;
PROCEDURE SORTMERGESTMT(MERGE); VALUE MERGE; BOOLEAN MERGE; FORWARD;
BEGIN EFFSEGMENT;
PROCEDURE CONSTANTCLEANP;
%
BEGIN EFFSEGMENT;
END B2D;
$VOID 04277000
BEGIN EFFSEGMENT;
BUGGER(S.[36:10]) ELSE WOP[T];IF T=1 THEN RR10
BUGGER(S.[36:10]) ELSE WOP[T];IF T=1 THEN RR10
END EFFSEGMENT DEBUG;
COMMENT PACK IS A STREAM PROCEDURE WHICH INSERTS THE SYLLABLE
INTO THE EDOC ARRAY, THE SPECIFIC ELEMENT OF EDOC
IS PRECISLY = EDOC[(L DIV 4) DIV 128+(L DIV 4)MOD 128]
SYLLABLE POSITION=(L MOD 4), WHERE L IS THE SYLLABLE
NUMBER RELATIVE TO THE BEGINNING OF THE SEGMENT;
STREAM PROCEDURE PACK(WORD,POSITION,SYLLABLE);
VALUE POSITION,SYLLABLE;

```

```

09419500
09419750
13681100
09361005
09161040
09361050
09161080
09361100
13113000
413148200
01800060
01339050
01340500
01341000
01416500
01525000
01583000
01598000
01684000
01537000
01786000
02801010
02801610
02013184
02020000
02262000
02562000
02691000
02716000
02904000
03018000
03031000
03034000
03034100
03874000
03081000
03082000
03082100
04143000
04163000
04164000
04247090
04248010
04265000
04276100
04281000
04286000
04287100
04292100
04292110
04292120
04292130
04292140
04292150
04292160

```

```

BEGIN
DI=WORD ; DI ← DI+POSITION ; DI ← DI+POSITION;
SI←LOC SYLLABLE ; SI←SI+6;
DS←2 CHR ;
END PACK ;
RR10:= " " ; %
RR10:=(IF Q=10 THEN OP.[39:3]&OP[41:38:1] ELSE
DEFINE PUNCH(FL,ST)=IF WRITNEW(FL,ST) THEN#;
$ VOID 05056001
$VOID 05101000
BEGIN EFFSEGMENT; FLAG(ERRNUM);
EFFSEGMENT;
BOOLEAN RRB2;
DEFINE CHECKPRESENCE = % FOR EFF
END CHECKPRESENCE;#;
BEGIN EFFSEGMENT;IF SPAC=0 THEN SPAC←GETSPACE(TRUE,-2);
EFFSEGMENT;
BEGIN EFFSEGMENT; RANGE←ELCLASS≥LOWER AND ELCLASS≤UPPER END;
DEFINE CALLSWITCH(H)= %
#;
$ VOID 05325001
$ VOID 05387001
$VOID 05407001
DEFINE IFCLAUSE = %
BEGIN STEPIT; STACKCT ← 0; BEXP;
IF ELCLASS ≠ THENV THEN ERR(116)ELSE STEPIT END IFCLAUSE;
#;
$ VOID 06411001
EFFSEGMENT;
EFFSEGMENT;
BEGIN
HOLE ← 2 - REAL(FROM);
VARIABLE(HOLE);
GO TO EXIT
END;
EFFSEGMENT;
DEFINE C=RR1#; % TO HOLD L FOR A WHILE
BEGIN
DEFINE C=N#;
BEGIN
$ VOID
I1=I+1; STEPIT; %
MOVECODE(TEDOC,EDOC); %
$ VOID 08276001
LOCK(FILEATTRIBUTES[*]); I:=FILEATTRIBUTES[0];
%
PROCEDURE SORTMERGESTMT(MERGE); VALUE MERGE; BOOLEAN MERGE;
BEGIN INTEGER J,K,FILEL,FILEND; LABEL QUIT; BOOLEAN OPTOG,INPRO;
DEFINE A=K#; RDS=1,280#; OUTPRO=OPTOG#;
COMMENT ARRAYCHECK CHECKS A PARAMETER=INFO WORD FOR SORT/MERGE;
BOOLEAN PROCEDURE ARRAYCHECK(AAW); VALUE AAW; REAL AAW;
ARRAYCHECK←AAW,CLASS<BOOARRAYID OR AAW,CLASS>INTARRAYID
OR AAW,INCR #I;
COMMENT COMMACHECK LOOKS FOR COMMAS AND STEPS AROUND THEM;
BOOLEAN PROCEDURE COMMACHECK;
BEGIN IF NOT(COMMACHECK←(STEP1=COMMA)) THEN ERR(350);
STEPIT

```

```

04292170
04292180
04292190
04292200
04292210
04602000
04406000
05831100
05050000
05061000
05106000
05152500
05166100
05184000
05187000
05760000
05293100
05104000
05315000
05316100
05317000
05379000
05388000
06291100
%A 06291200
06291300
06291400
06409000
06414100
06420100
07402000
07402100
07402200
07402300
07402400
07538100
07596100
07609200
07646397
07646760
07689000
07691000
07698500
08246000
08493140
08901000
08901200
08902000
08902010
08902020
08902030
08902040
08902050
08902060
08902070
08902080
08902090

```

```

%ELIMINATE %74
%NAME CALL %74
%EXPRESSION %74
%PRT CELL %74
%ACC ENTRY %74

```

MODULE BUSINESS SYSTEMS, INC., NY

```

END COMMACHECK;                                08902100
COMMENT HVCHECK CHECKS VALIDITY OF HIVALU PROCEDURE FOR SORT; 08902200
BOOLEAN PROCEDURE HVCHECK(ELBW); VALUE ELBW; REAL ELBW; 08902210
IF ELBW.CLASS#PROCID THEN ERR(356) ELSE 08902220
IF BOOLEAN(ELBW.FORMAL) THEN HVCHECK*TRUE ELSE 08902230
IF TAKE(GT1+GIT(ELBW))#1 THEN ERR(357) ELSE 08902240
IF ARRAYCHECK(TAKE(GT1+1)) THEN ERR(358) ELSE 08902250
HVCHECK*TRUE; 08902260
COMMENT OUTPROCHECK CHECKS SORT/MERGE OUTPUT PROCEDURE; 08902270
BOOLEAN PROCEDURE OUTPROCHECK(ELBW); VALUE ELBW; REAL ELBW; 08902280
IF ELBW.CLASS#PROCID THEN ERR(351) ELSE 08902290
IF BOOLEAN(ELBW.FORMAL) THEN OUTPROCHECK*TRUE ELSE 08902300
IF TAKE(GT1+GIT(ELBW))#2 THEN ERR(352) ELSE 08902310
IF TAKE(GT1+1).CLASS#BOOID THEN ERR(353) ELSE 08902320
IF ARRAYCHECK(TAKE(GT1+2)) THEN ERR(354) ELSE 08902330
OUTPROCHECK*TRUE; 08902340
COMMENT EQLESHECK CHECKS THE COMPARE ROUTINE FOR SORT/MERGE; 08902350
BOOLEAN PROCEDURE EQLESHECK(ELBW); VALUE ELBW; REAL ELBW; 08902360
IF ELBW.CLASS#BOOPROCID THEN ERR(359) ELSE 08902370
IF BOOLEAN(ELBW.FORMAL) THEN EQLESHECK*TRUE ELSE 08902380
IF TAKE(GT1+GIT(ELBW))#2 THEN ERR(360) ELSE 08902390
IF ARRAYCHECK(TAKE(GT1+1)) THEN ERR(361) ELSE 08902400
IF ARRAYCHECK(TAKE(GT1+2)) THEN ERR(362) ELSE 08902410
EQLESHECK*TRUE; 08902420
STREAM PROCEDURE STUFFILE(IDLOC*FN, SFN); 08902430
VALUE FN, SFN; 08902440
BEGIN DI-IDLOC; DI+DI+5; DI+DC; 08902450
SI+LOC FN; SI+SI+5; DS+3 CHR; SI+SI+7; 08902460
DS+11 LIT"0000000DSRT"; DS+CHR; SI+SI-1; 08902470
DS+7 LIT"SDSRT"; DS+CHR; SFN+DI; SI+LOC SFN; 08902480
DI-IDLOC; DI+DI+5; SI+SI+5; DS+3 CHR; 08902490
END STUFFILE; 08902500
BOOLEAN PROCEDURE INPROCHECK(ELBW); VALUE ELBW; REAL ELBW; 08902510
IF ELBW.CLASS#BOOPROCID THEN ERR(363) ELSE 08902520
IF BOOLEAN(ELBW.FORMAL) THEN INPROCHECK*TRUE ELSE 08902530
IF TAKE(GT1+GIT(ELBW))#1 THEN ERR(364) ELSE 08902540
IF ARRAYCHECK(TAKE(GT1+1)) THEN ERR(365) ELSE 08902550
INPROCHECK*TRUE; 08902560
IF MERGE THEN 08903000
BEGIN % MERGE 08904000
END MERGE 08926000
ELSE 08927000
BEGIN % SORT 08928000
$VOID 08945000 08929000
END SORT; 08996100
END SORT AND MERGE STATEMENTS; 08998000
$ VOID 09024000 09003000
$ VOIDT 09024000
PROCEDURE GETREADY; 09025000
BEGIN EFFSEGMENT; 09025100
$ VOIDT 09027000
$ VOIDT 09028073 09028068
$ VOID 09056001 09042000
LOCK(TENT[*]); % READ ONLY 09251310
LOCK(SPECIAL[*]); % READ ONLY 09251320
LOCK(SUPERSTACK[*]); % READ ONLY 09251350
GT1:=GETSPACE(TRUE,-5); ERRORTOGT*TRUE; 09774000
END GETTING READY; 09774010

```

```

PROCEDURE CLEANUP;                                09274020
  BEGIN LABEL LI;                                  09274030
  STREAM PROCEDURE MDESC(WD,TOLOC);VALUE WD;      09274040
  BEGIN DI←LOC WD; DS← SET;SI← LOC WD; DI←TOLOC;DS←WDS END; 09274050
  COMMENT THE FOLLOWING PROCEDURE PRINTS OUT THE PRT, NAME, AND 09274060
  SEGMENT NUMBER OF THE INTRINSIC PROCEDURES USED IN THE 09274070
  OBJECT PROGRAM;                                  09274080
  STREAM PROCEDURE WRTINTRSC(SGNO,ALFA,PRT,FIL);  09274090
  VALUE SGNO,PRT;                                  09274100
  BEGIN LOCAL COUNT,DEST;                          09274110
  DI←FIL; DS←4 LIT"PRT("); SI←LOC PRT; SI←SI+4; TALLY←4; 09274120
  3(IF SC="0" THEN % DONT PRINT LEADING ZEROES, 09274130
  BEGIN SI←SI+1; TALLY←TALLY+63 END ELSE JUMP OUT); 09274140
  COUNT←TALLY; DS←COUNT CHR; DS←4 LIT") = "; 09274150
  SI←ALFA; SI←SI+2; DEST←DI; % SAVE DI, 09274160
  DI←LOC COUNT; DS←7 LIT"0"; DS←CHR; % NO OF CHARS IN NAME, 09274170
  DI←DEST; DS←COUNT CHR; % INT, NAME, 09274180
  DS←29 LIT" INTRINSIC, SEGMENT NUMBER = "; 09274190
  SI←LOC SGNO; DS←4 DEC; DS←LIT","; 09274200
  DI←DI-5; DS←4 FILL; % JUNK LEADING BLANKS, 09274210
  END WRTINTRSC;                                  09274220
  DEFINE STARTINTRSC=426#;                          09274230
  ARRAY PRT,SEGDICT[0:7,0:127];                      09274240
  INTEGER PRTADR,SEGMNT,LINK;                        09274250
  INTEGER PROCEDURE MOVEANDBLOCK(FROM,SIZE); VALUE SIZE; 09274260
  ARRAY FROM[0,0]; INTEGER SIZE;                    09274270
  BEGIN INTEGER NSEGS,I,J;                            09274280
  REAL T;                                              09274290
  NSEGS←(SIZE+29) DIV 30;                             09274300
  IF DA DIV CHUNK<T+(DA+NSEGS) DIV CHUNK THEN 09274310
  DA←CHUNK×T;                                          09274320
  MOVEANDBLOCK←DA;                                    09274330
  DO BEGIN FOR J←0 STEP 2 WHILE J<30 AND I<SIZE DO 09274340
  BEGIN MOVE(2,FROM[I DIV 128,I MOD 128],CODE(J)); 09274350
  I←I+2; END;                                         09274360
  WRITE(CODE[DA]); DA←DA+1;                            09274370
  END UNTIL I≥SIZE;                                    09274380
  END MOVEANDBLOCK;                                    09274390
  STREAM PROCEDURE ZEROUT(DEST,NDIV32,NMOD32); 09275000
  VALUE NDIV32,NMOD32 ;                               09275010
  BEGIN DI←DEST;                                       09275020
  NDIV32(32(DS ←8 LIT"0"));                             09275030
  NMOD32(DS ← 8 LIT"0"));                               09275040
  END;                                                  09275050
  $ VOID 09333000                                       09317000
  $ VOID 09348000                                       09334000
  GTI←MIN((IDLOC-MKABS(IDARRAY[0])),(33+15)+1,128); % 09394000
  BEGIN                                                09409000
  $ VOID 09415000                                       09410000
  END OF CLEAN UP;                                     09420000
  %%%%%%%%%%% ACTUAL COMPILATION DONE HERE %%%%%%%%%%% 09421000
  DEFINE PROGRAM =                                     09421100
  GETREADY;                                           09422000
  BLOCK(FALSE);                                       09423000
  CLEANUP;                                           09424000
  *;                                                  09424100
  %%%%%%%%%%%                                           09425000

```

Moore Business Forms, Inc. NY


```

%
EFFSEGMENT;
COMMENT GETINT DOES A CALL ON NEXTENT AND CHECKS TO SEE IF AN INTEGER
WAS THE RESULT; IF NOT ERROR = OTHERWISE MAKE SIGN PLUS;
BOOLEAN PROCEDURE GETINT;
BEGIN NEXTENT; IF ELCLASS * = ELCLASS < 0 THEN
IF ELCLASS = ("**") THEN GETINT = TRUE ELSE
BEGIN FLAG(137); ELCLASS * 0 END
END GETINT;
COMMENT DIVIDE PARTIONS THE PARAMETER NUMBER INTO THREE PARTS, THE
RESULT IS PASSED BACK THROUGH P1, P2, AND THE FUNCTION
IDENTIFIER, SEE CODE FOR DETAILS;
INTEGER PROCEDURE DIVIDE(NUMBER, P1, P2);
VALUE NUMBER; INTEGER P1, P2, NUMBER;
BEGIN
IF NUMBER < 0 THEN BEGIN FLAG(138); ERRORTOG = TRUE;
NUMBER * 0 END;
P1 * IF NUMBER < 8 THEN NUMBER ELSE 8;
NUMBER * NUMBER / P1;
P2 * IF NUMBER < 8 THEN NUMBER ELSE 8;
DIVIDE * NUMBER / P2 END DIVIDE;
$ VOID 10193000
DEFINE PACKINFO(INFO, ISKIP, COUNT, ASKIP, ACCUM) =
MOVECHARACTERS(COUNT, ACCUM, ASKIP + 3, INFO, ISKIP) #;
$ VOID 10240001
$ VOID 13144001
$ VOID 13210001
BEGIN EFFSEGMENT;
$ VOID 13537001
PUT(X, NEXTINFO); % FOR EFF...
BEGIN EFFSEGMENT;
DEFINE MOVE2(S, D) = MOVE(2, S, D) #;
$ VOID 13676001
$ VOID 13682001
EFFSEGMENT;
BEGIN HANDLESWLST; %SAVE
GOTSTORAGE * NOT SPECTOG OR GOTSTORAGE; %PRI
GO TO START END; %CELL
DEFINE CHRJOB = IF GTAT(J) = J * 1] # 0 THEN FLAG(23) #; %
$ VOID 14273201
EFFSEGMENT;
EFFSEGMENT;
$ VOID 15019001
REAL STREAM PROCEDURE GETALPHA(INFOINDEX, SIZE);
VALUE SIZE I
BEGIN COMMENT GETALPHA PICKS ALPHA CHARACTERS OUT OF INFO AND
FORMATS THE ID WORD THAT IS PASSED TO PRINTI, THE FIRST
CHARACTER CONTAINS THE SIZE, THE NEXT CHARACTER CONTAINS THE
ALPHA LEFT JUSTIFIED WITH TRAILING ZEROS;
DI = LOC GETALPHA; DS = 8 LIT "0 "; DI = DI - 7;
SI = INFOINDEX; SI = SI + 3; DS = SIZE CHR;
END GETALPHA;
$# PATCH NUMBER 406 FOR XALGOL CONTAINS 31 CARDS
$! THIS MAKES 8 CHARACTER BCL, 16 OCTADE OCTAL AND 12 HEXADE HEX STRINGS
$! ACCEPTABLE SO LONG AS THE FLAG BIT IS NOT SET, 5/15/72 DHB
$! THIS CHANGES THE WAY THE COMPILER TREATS OCTAL AND HEX STRINGS,
$! THEY ARE NOW CONSIDERED AS A STRING OR STRINGCON AS OTHER STRINGS
$! THIS DOES NOT CHANGE THE WAY ALGOL USES THESE CONSTANTS BUT

```

```

09426000
10070000
10105005
10105010
10105020
10105030
10105040
10105050
10105060
10105070
10105080
10105090
10105100
10105110
10105120
10105130
10105140
10105150
10105160
10105170
10105180
10175000
10733100
10233200
10236000
13115000
13209000
13369000
13532000
13593000
13632100
13474000
13675000
13682000
13715100
14168000
14168100
14168200
14773100
14273200
14527000
15006100
15014000
15830100
15030200
15830300
15030400
15030500
15030600
15030700
15030800
15030900
09417110
02776030
02847005
02847010
02847015

```

```

$! IN XALGOL THE STATEMENTS:
$!   REPLAGE . . . BY "A";   AND
$!   REPLAGE . . . BY 3"21";
$! WILL GENERATE THE SAME RESULT,   1/16/74.
PROCEDURE DOOCTALORHEXSTRING;
BEGIN LABEL AWAY; %
    RESULT:=5; SCANNER; % SKIP QUOTE,
    COUNT:=0; GT1:= 48 DIV C; % 16 OR 12
    DO BEGIN
        RESULT:=5; SCANNER;
        IF COUNT > GT1 THEN % > 1 WORD LONG,
            BEGIN ERR(520); GT1:=0; GO AWAY END;
        END UNTIL EXAMIN(NCR)="";
        Q:=ACCUM[1]; RESULT:=5; SCANNER; COUNT:=COUNT-1;
        IF C=3 THEN % OCTAL STRING,
            IF OCTIZE(ACCUM[1],ACCUM[4],16-COUNT,COUNT) THEN
                FLAG(521); % NON-OCTAL CHARACTER IN STRING,
            ELSE ELSE IF HEXIZE(ACCUM[1],ACCUM[4],12-COUNT,COUNT) THEN
                FLAG(521); % NON-HEX CHARACTER IN HEX STRING,
            COUNT:=(COUNT*C+C) DIV 6; % NUMBER OF 6 BIT CHAR IN STRING
            MOVECHARACTERS(COUNT,ACCUM[4],8-COUNT,ACCUM[1],3);
            GT1:=1; Q:=ACCUM[1];=(*)&COUNT[12;42;6];
            AWAY; %
    END OCTAL OR HEX STRING;
    STRNGXT,ARGH; %
    IF COUNT < 8 OR
        COUNT=8 AND NOT BOOLEAN(Q,[18;1]) THEN % NO FBIT STILL OK
    $ VOID 02704001
    BEGIN %
        DOOCTALORHEXSTRING; %
        IF BOOLEAN(GT1) THEN GO TO STRNGXT; % WAS VALID
    $ VOID 02765001
        GO TO SCANAGAIN; % WAS AN ERROR
    $ VOIDT 02782001
    $ VOID 02847001
    $# PATCH 407 FOR XALGOL.XIII CONTAINS 22 CARDS
    IF ELCLASS=CROSSHATCH THEN BEGIN
        STEPIT; STMPARITH END ELSE
        INTEGER SL;
        IF ELCLASS=FACTOR THEN
            IF LYNKTG THEN % LAST SYLLABLE IS NOT A LINK
                IF (SL*GET(L=1)), [46;2]=3 OR SL=673 THEN BEGIN
                    ELBAT[1],CLASS+ELCLASS+CROSSHATCH; STEPIT;
                    EMIT(DUP); EMIT(COD); GO EXIT END;
                IF ELCLASS=CROSSHATCH THEN GT4+BOOSEC+BTYPE ELSE
                    IF P1 # FL THEN STEPIT;
                    IF T1#0 OR ELCLASS=FACTOR THEN BEGIN EMIT(DUP);
                        EMIT(COC); IF T1#0 AND ELCLASS=FACTOR THEN
                            EMIT(DUP) END;
                    ELSE IF T1#0 OR ELCLASS=FACTOR THEN BEGIN
                        EMITV(TALL ) ; IF T1#0 AND ELCLASS=FACTOR
                            THEN EMIT(DUP) END;
                IF ELCLASS=FACTOR THEN BEGIN %93
                    ELBAT[1],CLASS + ELCLASS +CROSSMATCH END; %93
                BEGIN%
                IF ELCLASS=FACTOR THEN BEGIN
                    ELBAT[1],CLASS+ELCLASS+CROSSHATCH;

```

```

02847020
02847025
02847030
02001855
02001856
02001900
02001905
02001910
02001915
02001920
02001925
02001930
02001935
02001940
02001945
02001950
02001955
02001960
02001965
02001980
02801985
02001990
02801995
02438000
02503000
02703010
02704000
02562000
02763000
02764000
02765000
02766000
02767000
02847000
06003200
06003400
06089500
06091200
06091300
06091400
06091500
06091600
06152500
15092015
15095000
15095500
15095600
15097000
15097500
15097600
15098025
15098050
15098200
15306200
15306400

```

```

EMIT0(DUP); IF T1=0 THEN EMIT0(LOD) END; 15306600
S# PATCH 408 FOR XALGOL.XIII CONTAINS 19 CARDS
LABEL EXIT,GETC; % 08017000
OR ELCLASS=LEFTPAREN OR ELCLASS=COLON 08103100
IF ELCLASS=COLON THEN % 08111100
BEGIN% 08111200
SIMPLEBI=CONSTANBI=TRUE; SIGNBI=FALSE;% 08111300
BI=1; ELCLASS:=UNTILV;% 08111400
PLUG(TRUE,1);% 08111500
END% 08111600
ELSE% 08111700
(STEPI=UNTILV OR ELCLASS=RTTPAREN OR ELCLASS=WHILEV) 08113000
OR ELCLASS=RTTPAREN % 08128100
BEGIN %% COLON MAYBE 08182100
IF ELCLASS=COLON THEN% 08182200
BEGIN% 08182300
CONSTANBI=TRUE; SIGNBI=FALSE; BI=1; GO GETC % 08182400
END;% 08182500
IF FORCLASS(UNTILV) OR ELCLASS=RTTPAREN THEN% 08184000
GETC; IF SIMPLECONSTANC,@,SIGNC) THEN% 08185000
END; %% MAYBE COLON 08722100
S# PATCH NUMBER 410 FOR XALGOL CONTAINS 389 CARDS
S! IMPLIMENTS SYSTEMID, NABS AND NOBITSON, 10/26/72
S! THIS IS THE IPC PATCH FOR ALGOL AND XALGOL. D, H, BROWN---JAN, 1973
TASKID =115#, COMMENT 163; 01778060
EVENTID =116#, COMMENT 164; 01778065
IPCPROCID =117#, COMMENT 165; 01778070
SIZEV =21#, COMMENT 25; 01298600
TASKV =22#, COMMENT 26; 01298650
EVENTV =23#; COMMENT 27; 01298700
BOOLEAN IPCJOB; % TRUE IF "PROCEDURE" IS FIRST THING TO BE COMPILED, 01401100
% I.E. IF WE ARE COMPILING AN INVOKABLE IPC PROGRAM, 01401110
% AFTER PROCESSING THE IPC PROCEDURE SPECIFICATION THE 01401120
% CF CONTAINS 2*MARK, WHICH POINTS TO THE PARAMETER 01401130
% SPECIFICATIONS IN INFO, THE FF IS USED TO COUNT THE 01401140
% NUMBER OF "PROCESS" STATEMENTS, IF IPCJOB IS NON ZERO 01401150
% AT WRAP UP THEN THE CODE FILE IS MARKED AS INVOKING 01401160
% AND/OR INVOKED AND, IF NEEDED, A PARAMETER DESCRIPTION 01401170
% SEGMENT IS WRITTEN IN THE CODE FILE, 12/13/72 DHBROWN 01401180
POWERSOFTEN =758#, 01574000
LASTSEQUENCE =249#, % 760 = 511 01575000
SORTA =759#; 01580000
<X21,"CUSTOMS B=5700 IPC XALGOL COMPILER MARK ", 01830000
PROCEDURE TASKATTRIBUTEHANDLER(F); VALUE F; INTEGER F; FORWARD;%0776600003061250
MAXANDMIN, DELAY,DELTA, OTHERS, SYSTEMID, EXIT; 06060100
MAXANDMIN, MAXANDMIN, DELAY, ABS, % NABS 06061100
SYSTEMID; 06061150
IF BOOLEAN((T1:=ELBAT[I]),VO) THEN % WRONG TYPE BOOLEAN 06061500
BEGIN ERR(103); GO TO EXIT END; 06061750
IF T2:=T1.[27+6]<9 OR T2=50 THEN 06862000
GO TO S; IF T2=50 THEN T2=40 ELSE T2+1; 06062050
ABS; PAN; EMIT0(IF T2=50 THEN SSN ELSE SSP); GO TO EXIT; 06067000
SYSTEMID; EMITL(7); EMITL(1); EMIT0(COM); STEPIT; GO EXITT; % FPM 06073150
IF ELCLASS=EVENTID AND REALF THEN % REAL(EVENTID) IS OK 06090178
BEGIN EMITV(ELBAT[I]); STEPIT; GO EXIT END; % EVENTS ON STACK 06090179
IF ELCLASS=TASKID THEN % 06092075
BEGIN TASKATTRIBUTEHANDLER(F); GO TO LAMPER END; 06092077

```

```

PROCEDURE BIMPFUN; % HANDLES THE BOOLEAN SPECIAL FUNCTIONS          06210010
BEGIN %                                                              06210020
  LABEL OTHERS, FIX, FREE, AVAILABLE, ACQUIRE, HAPPENED, PEND, NOBITSON, 06210030
  EXIT, CKRPN;                                                       06210040
  SWITCH %                                                            06210050
  FUN:=OTHERS, FIX, FREE, AVAILABLE, ACQUIRE, HAPPENED, PEND, NOBITSON; 06210060
  DEFINE HAPBIT = 8#; % BIT [44:1] IS ON IF EVENT HAS HAPPENED,      06210070
  AVLBIT = 1#; % BIT [47:1] IS ON IF EVENT IS NOT AVAILABLE,        06210080
  % IF THE AVLBIT IS ON THEN EVENT, FF CONTAINS                      06210090
  % THE MIX INDEX OF THE PROCESS IN CONTROL.                         06210100
  REAL T1, T2;                                                       06210110
  %                                                                    06210120
  %                                                                    06210130
  IF I ≤ T1:=ELBAT[I], [27:6] ≤ 6 THEN % FIRST PARAMETER MUST BE EVENT 06210140
  BEGIN                                                              06210150
    IF STEP1 ≠ LEFTPAREN THEN BEGIN ERR(105); GO EXIT END;          06210160
    IF STEP1 ≠ EVENTID THEN BEGIN ERR(554); GO EXIT END;           06210170
    T2:=ELBAT[I]; % THE EVENT                                        06210180
    IF REAL(IPCJOB), [33:15] ≠ 0 THEN % COMPILING AN INVOKED IPC JOB 06210190
    IF T2, [9:7] = 65 AND T1 ≤ 2 THEN % CANT FIX OR FREE A          06210200
    FLAG(555); % FORMAL LEVEL 1 EVENT                               06210210
    STEPIT;                                                         06210220
  END;                                                                06210230
  GO TO FUN[T1+1]; % TO PROPER SYNTAXER                              06210240
  OTHERS: GO EXIT; % ACTUAL BOOLEAN INTRINSICS WOULD GO HERE        06210250
  FIX: % MAKE THE EVENT ***NOT*** AVAILABLE AND RETURN FORMER STATE 06210260
  FREE: % MAKE THE EVENT *** AVAILABLE *** AND RETURN FORMER STATE 06210270
  EMITV(T2); EMITO(DUP); % 2 COPIES ON STACK                        06210280
  EMITPAIR(HAPBIT, LND); % RESETS EVERYTHING BUT HAPBIT            06210290
  IF T1=1 THEN EMITPAIR(AVLBIT, LOR); % FIX = BIT ON SO NOT AVAIL 06210300
  EMITN(T2); EMITO(STD); % CHANGE IT                                06210320
  EMITO(LNG); % RETURN TRUE IF IT WAS AVAILABLE,                    06210330
  GO CKRPN; %                                                       06210340
  AVAILABLE: % RETURN TRUE IF THE EVENT IS AVAILABLE (AVL BIT OFF) 06210350
  EMITV(T2); EMITO(LNG); GO CKRPN;                                  06210360
  ACQUIRE: % WAIT FOR AVLBIT OR TIME OUT                            06210370
  PEND: % WAIT TILL HAPBIT OR TIME OUT                              06210380
  EMITN(T2); % ADDRESS OF THE EVENT                                 06210390
  EMITL(IF T1=4 THEN AVLBIT ELSE HAPBIT);                           06210400
  IF ELCLASS≠COMMA THEN BEGIN ERR(556); GO EXIT END;               06210410
  STEPIT; AEXP; % TIME OUT IN SECONDS                              06210420
  EMITPAIR(44, COM); % GO DO IT                                    06210440
  EMITO(DEL); EMITO(DEL); % LEAVE RESULT OF FUNCTION ON STACK     06210450
  GO CKRPN;                                                         06210460
  HAPPENED: % RETURNS TRUE IF THE EVENT HAS THE HAPPENED BIT ON    06210470
  EMITV(T2); EMITPAIR(HAPBIT, LND); EMITPAIR(HAPBIT, EQL);        06210480
  GO CKRPN; %                                                       06210490
  NOBITSON: % NOBITSON(X) = NOT X = NOT 0                            06210500
  PAN: EMITO(LNG); EMITPAIR(0, LNG); EMITO(EQL);                   06210510
  GO EXIT;                                                         06210520
  CKRPN: IF ELCLASS ≠ RTPAREN THEN ERR(104) ELSE STEPIT;           06210530
  EXIT;                                                             06210540
  END BOOLEAN IMPLIED FUNCTIONS;                                     06210550
  IF ELCLASS≠TASKID THEN GO TO L33; % ALWAYS ARITHMETIC            06227550
  GO TO L18; % FOR SAFETY ONLY                                     06234500
  L11: IF BOOLEAN(ELBAT[I], VO) THEN % BOOLEAN INTRINSIC           06235000
  BEGIN BIMPFUN; GO TO TD END; % ALLOW , AND & AFTER IT           06235500

```

```

L18: L19: L20: L22: L23: L24: L26: L27: L28: L31: L32: L33:      06236000
  IF ACLASS=TASKID THEN GO TO LODPOINT; % NAME CALL ON TASKID 07217515
  IF ACLASS=EVENTID THEN GO TO L22; % NAME OR VALUE CALL EVENTID 07217517
COMMENT  IF THE INTRINSIC IS ONLY IMPLIMENTED IN LINE EMIT AN ERR; 07315100
  IF TAKE(WHOLE+WHOLE,[33:2]) < 0 THEN FLAG(552); 07315200
  IF ELCLASS=TASKID THEN 07529300
    BEGIN TASKATTRIBUTEHANDLER(FS); GO EXIT END; 07729400
PROCEDURE IPCSTATEMENT(W); VALUE W; INTEGER W; 07560000
  BEGIN % VALUE OF W PASSED 1 2 3 4 5 07560100
    LABEL EXIT, LIBERATE,PROCURE,WAIT,CAUSE,RESET; 07560200
    SWITCH STMT:= LIBERATE,PROCURE,WAIT,CAUSE,RESET; 07560300
    DEFINE HAPBIT=#; % BIT [44:1] IS ON IF THE EVENT HAS HAPPENED 07560400
    AVLBIT=#; % BIT [47:1] IS ON IF THEN EVENT IS NOT AVAILIABLE 07560500
    REAL E; 07760600
    IF STEPI # LEFTPAREN THEN BEGIN ERR(105); GO EXIT END; 07760700
    IF STEPI # EVENTID THEN BEGIN ERR(557); GO EXIT END; 07760800
    E:=ELBAT[I]; 07560900
    GO TO STMT[W]; 07561000
    LIBERATE: % CALL COM 44 PASSING: EVENT ADDR, 0, NOT 0 - RESETS AVLBIT 07561100
    PROCURE: % CALL COM 44 PASSING: EVENT ADDR, 1, NOT 0 - GETS CONTROL 07561200
    WAIT: % CALL COM 44 PASSING: EVENT ADDR, 8, NOT 0 - SLEEPS TILL HAP 07561300
    EMITN(E); 07561400
    EMITL(IF W=1 THEN 0 ELSE IF W=2 THEN AVLBIT ELSE HAPBIT); 07561500
    EMITL(0); IF W=1 THEN EMITO(LNG); EMITPAIR(44,COM); 07561600
    EMITO(DEL); EMITO(DEL); EMITO(DEL); % CUT BACK STACK 07561700
    GO EXIT; 07561800
    CAUSE: % TURN ON HAPPENED BIT 07561900
    RESET: % TURN OFF HAPPENED BIT 07562000
    EMITV(E); 07562100
    EMITPAIR(HAPBIT,IF W=4 THEN LOR ELSE LNG); 07562200
    IF W=5 THEN EMITO(LND); 07762300
    EMITN(E); EMITO(STD); 07562400
    EXIT: IF STEPI # RTPAREN THEN ERR(104) ELSE STEPI; 07562500
  END IPC STATEMENT; 07562600
PROCEDURE PROCESSTATEMENT; 07565000
  BEGIN % HANDLE: PROCESS IPCEXTERNAL(AP1,AP2,AP3)[TASKID]; 07565025
    REAL E,T; 07765050
    INTEGER N,MARK,PCNT; 07765075
    LABEL BAD,Z; 07565100
    DEFINE FAULT(F) = BEGIN T:=F; GO BAD END#, FF=18:15#; 07565125
    IPCJOB,[FF]:=BOOLEAN(REAL(IPCJOB,[FF])+1); % COUNT PROCESS STMNTS 07565150
    IF STEPI # IPCPROCID THEN FAULT(559); % MUST BE EXTERNAL PROCEDURE 07765175
    SCATTERELBAT; 07565200
    EMITO(MKS); % MARK STACK FOR CHAR MODE CUT BACK WITH XITR AFTER COM 07565250
    IF NI=TAKE(MARK:=LINKF+INCRF)=0 THEN GO TO Z; % # PARAMETERS 07565275
    IF STEPI # LEFTPAREN THEN FAULT(105); 07765300
    DO BEGIN 07765325
      IF BOOLEAN((T:=TAKE(MARK+PCNT:=PCNT+1)),VO) THEN % VALUE 07565350
        BEGIN STEPIT; % 07565375
          IF EXPRSS#(IF T,CLASS=BOOID THEN BTYPE ELSE ATYPE) THEN 07765380
            FAULT(123); T:=3; 07765385
        END % 07765390
      ELSE % 07765400
        BEGIN % NAME CALL 07565425
          IF STEPI # T,CLASS THEN FAULT(123); E:=ELBAT[I]; 07565450
          IF BOOARRAYID ≤ ELCLASS ≤ INTARRAYID THEN 07565475
            BEGIN EMITPAIR(E,LOD); T:=4; 07765500
              DO UNTIL STEPI=COMMA OR ELCLASS=RTPAREN; 07765525

```

```

END
ELSE
  BEGIN EMITN(E);
    T1=IF ELCLASS=EVENTID THEN 1 ELSE 2; STEPIT;
  END;
END NAME CALLS;
EMITL(T); % PARAMETER TYPE
END
UNTIL ELCLASS # COMMA OR PCNT=N;
IF PCNT # N THEN FAULT(128);
IF ELCLASS # RTPAREN THEN FAULT(104);
Z: IF STEP1 # LFTBRKET THEN FAULT(560);
EMITL(N); % NUMBER OF PARAMETERS
IF STEP1 # TASKID THEN FAULT(560);
EMITPAIR(ELBAT[1],LOD);
IF STEP1 # RTBRKET THEN FAULT(560) ELSE STEPIT;
EMITPAIR(42,COM); % GO DO IT
EMITN(X1TR); % CALLS FIRST SYL OF SEG 1 =EXC= TO CUT BACK STACK
IF FALSE THEN
BAD: ERR(T);
END PROCESS STATEMENT;
PROCEDURE TASKATTRIBUTEHANDLER(F); VALUE F; INTEGER F;
COMMENT TASK ATTRIBUTE HANDLER PROCESSES ALL TASK ATTRIBUTES.
IT IS EXPLICITLY CALLED IN 2 PLACES:
1: IN STMT PASSING F = FS,
THIS HANDLES TASK ATTRIBUTE ASSIGNMENTS BY CALLING AEXP,
2: IN PRIMARY PASSING F = FP,
THIS ALWAYS LEAVES A VALUE ON THE STACK, IF AN ASSIGNMENT
IS REQUIRED IT IS MADE WITH A "SND". IF NOT THE CURRENT
VALUE OF THE ATTRIBUTE IS LEFT ON THE STACK,
ACTUAL PARAMETER PART AND BPRIMARY DIRECT TRAFFIC THRU THE
ABOVE ENTRY POINTS TO DO THEIR WORK,
END COMMENT;
BEGIN
REAL E; INTEGER ATNDX;
E:=ELBAT[1]; % TASK ID
IF STEP1 = PERIOD THEN
BEGIN
STEPIT;
IF ATNDX=
  IF 0 = "9TASKV" THEN 0 ELSE
  IF 0 = "4MFID0" THEN 1 ELSE
  IF 0 = "3FID00" THEN 2 ELSE
  IF 0 = "6STATU" THEN 3 ELSE
  IF 0 = "7STACK" THEN 4 ELSE 562
  =562
THEN FLAG(562); % UNRECOGNIZED ATTRIBUTE
IF STEP1 = ASSIGNOP THEN
BEGIN % SET ONE
IF ATNDX=4 THEN FLAG(563); % CANT BE CHANGED
STEPIT; AEXP;
IF ATNDX=3 THEN % TSK,STATUS REQUIRES A COM FOR POSSIBLE DS,
BEGIN EMITPAIR(E,LOD); EMITPAIR(45,COM);
EMITO(DEL); IF F#FP THEN EMITO(DEL); % CUT BACK STK
END %
ELSE % JUST CHANGE THE ATTRIBUTE
BEGIN EMITL(ATNDX); EMITN(E);

```

```

07765550
07565575
07765600
07765610
07765615
07765625
07565650
07765675
07765700
07565725
07765750
07765775
07565800
07765825
07565850
07765875
07765900
07565925
07565950
07765975
07765985
07566000
07766025
07766050
07766075
07566100
07766125
07566150
07766175
07566200
07566225
07766250
07766275
07766300
07566325
07566350
07766375
07766400
07566425
07566450
07566475
07566500
07566525
07566550
07566575
07766600
07566625
07566650
07566675
07766700
07566725
07766730
07566735
07766740
07566745
07766750
07566755

```

```

EMIT0(IF F=FP THEN SND ELSE STD);
END; %
END
ELSE
IF F=FS THEN FLAG(564) % MISSING I=
ELSE
BEGIN EMITL(ATNDX); EMITV(E) END;
END
ELSE ERR(561); % MISSING ".
END TASK ATTRIBUTE HANDLER;
N:= IF Q = "8LIBER" THEN 1 ELSE
IF Q = "7PROCU" THEN 2 ELSE
IF Q = "5CAUSE" THEN 4 ELSE
IF Q = "5RESET" THEN 5 ELSE 0;
IF N # 0 THEN BEGIN IPCSTATEMENT(N); GO XXX END;
IF Q = "7PROCE" THEN BEGIN PROCSTATEMENT; GO XXX END;
BEGIN % IF I+2 IS EVENTID THEN ITS IPC WAIT
IF TABLE(I+2)=EVENTID THEN
BEGIN IPCSTATEMENT(3); GO XXX END;
N:=1; C:=2;
END ELSE %
OCT0132000000140000, "3FIX00",
OCT01300000006240000, "4NABS0",
OCT01320000000240000, "4FREE0",
OCT01320000000640000, "4PEND0",
OCT01300000000040000, "3TAN00",
OCT01300000006360000, "8SYSTE", "MID00000",
OCT013200000080760000, "8NOBIT", "SON00000",
OCT01320000000460000, "7ACQUI", "RE000000",
OCT01320000000360000, "9AVAIL", "ABLE0000",
OCT01320000000560000, "8HAPPE", "NED00000",
OCT04300000260000000, "4TASK0",
OCT04300000270000000, "5EVENT",
OCT14400000250000000, "4SIZE0",
OCT00300000260000000, "4FAST0",
OCT00300000270000000, "4SLOW0",
OCT2000000000004050, COMMENT POWERS OF TEN ;
0, COMMENT SORTA ;
" " ; COMMENT LASTSEQUENCE, LASTSEOROW ;
692 STEP 3 UNTIL 728, 732 STEP 4 UNTIL 748, 750 STEP 2 UNTIL 756
DO UNTIL STEPI=BEGINV OR %
ELCLASS=DECLARATORS AND ELBAT(I),ADDRESS=PROCVI % IPC
IF ELCLASS#BEGINV THEN % COMPILE AN INVOKABLE IPC PROGRAM,
BEGIN
IPCJOB:=TRUE; I:=I-1; % BACK UP TO LOOK AT PROCEDURE
END;
IF REAL(IPCJOB) # 0 THEN % INVOKED OR INVOKING IPC
BEGIN
IF MARK:=REAL(IPCJOB),[33;15] DIV 2 # 0 THEN % INVOKED IPC
IF IDARRAY[8]:=N:=TAKE(MARK) # 0 THEN % HAS PARAMETERS
BEGIN % BUILD PARAMETER DESCRIPTION SEGMENT
L:=0;
DO PRT[0,L]:= %
IF ELCLASS=(TI#TAKE(MARK+LI#L+1)),CLASS
# EVENTID THEN 1 ELSE
IF BOOARRAYIDSELCLASS$INTARRAYID THEN 4 ELSE
IF BOOLEAN(T,VO) THEN 3 ELSE 2
UNTIL L=N;

```

```

07766760
07566775
07766800
07766825
07766850
07766875
07566900
07766925
05766950
07566975
07810200
07810210
07810220
07810230
07810240
07810250
07825000
07825100
07825200
07825300
07825400
871309214421
871609214422
871909214423
872209214424
872509214425
872809214426
873209214427
873609214428
874009214429
874409214430
874809214431
875009214432
875209214433
8754 09214435
8756 09214436
8758 09214437
8759 09214440
8760 09214445
09214513
09252000
09252025
09252100
09252200
09252300
09252400
09405500
09405505
09405510
09405515
09405520
09405525
09405530
09405535
09405540
09405545
09405550
09405555

```

```

IDARRAY[9]:=MOVEANDBLOCK(PRT,N+1); 09405560
END INVOKED IPC JOB; 09405565
IDARRAY[2],[3:2]:=2+REAL(MARK#0); 09405570
COMMENT SEG0[2],[3:1] = JAR[2],[6:1] = IPC JOB 09405575
SEG0[2],[4:1] = JAR[2],[7:1] = INVOKED IPC JOB; 09405580
END IPC SET UP; 09405585
IF BOOARRAYIDSTYPESINTARRAYID OR TYPE=TASKID THEN 13330550
IF (FORMALFI=PTOG) AND NOT P2 THEN 13342000
BEGIN 13342250
PJ:=PJ+1; % COUNT PARAMETERS 13342500
ADDRSF:=IF IPCJOB THEN GETSPACE(TRUE,1)+64*PJ % PRT 13342750
ELSE PJ % STACK 13343000
END 13343500
ELSE 13344000
IF STOPGSP THEN ADDRSF:=0 13344500
ELSE ADDRSF:=GETSPACE(P2,1); 13345000
IF BOOARRAYIDSTYPESINTARRAYID OR TYPE=TASKID AND NOT IPCJOB 13346000
GT1:=IF PTOG THEN IF IPCJOB THEN LEVEL ELSE LEVELF ELSE LEVEL; 13170000
$ VOID 13371001 13371000
PROCEDURE TASK; % DOES TASK DECLARATION 13595025
BEGIN 13595050
LABEL EXIT; 13595075
JUMPCHKX; % 13595100
IF GTA1[J-1]#0 THEN FLAG(25); % CANT USE OWN OR SAVE 13595125
13595150
TYPEV:=TASKID; 13595175
IF NOT SPECTOG THEN EMIT0(MKS); 13595200
ENTRYCTYPEV; % DOES ALL OF THE LIST 13595225
IF ELCLASS#SEMICOLON THEN FLAG(0); 13595250
IF SPECTOG THEN GO EXIT; 13595275
EMITL(5); EMITL(1); EMITL(GTA1[0]); % SIZE, # DIMENSIONS, # TASK 13595300
EMITPAIR(1,SSN); % ASK BLOCK CONTROL FOR SAVE TASK ARRAYS 13595325
EMITV(5); % CALL BLOCK CONTROL 13595350
EXIT; 13595375
END TASK ENTR; 13595400
IPCOMPLETE,TASKDEC,EVENTDEC, % I P C 13595425
AUXSIZERR, % BAD USE OF AUXMEM OR SIZE 14016700
PTRDEC,DEFINDEC,AUXSIZERR,AUXSIZERR,TASKDEC,EVENTDEC; 14016800
LPB=[36:12]#, PJFLD=[16:5]#, % 14021000
$ VOID 1412500; 14023000
IF SPECTOG THEN % DOING A PROCEDURE SPECIFICATION 14069999
IF BUP#PJ THEN % ALL SPECIFICATIONS PROCESSED, DO CLEAN UP, 14870000
BEGIN 14871000
FI=FZERO; % 3*1600 = F+0, PARAMETER BASE ADDRESS 14072000
DO BEGIN % LOOK THRU PARAMS LAST TO FIRST & PUT IN PROC+INFO 14873000
G:=TAKE(BUP:=LASTINFO); % ELBAT OF LAST PARAM SPECIFIED 14074000
WHILE PJ#(IF IPCJOB THEN G,PJFLD ELSE G,ADDRESS) 14075000
DO G:=TAKE(BUP:=BUP-TAKE(BUP+1),PURPT); % GET PJ=TH ONE 14076000
TYPEV:=G,CLASS; 14077000
IF IPCJOB THEN 14078000
G,PJFLD:=0 % 14079000
ELSE 14081000
IF TYPEV#FRMTID OR TYPEV#SUPERFRMTID THEN 14082000
G,ADDRESS:=FI:=F+2 % LEAVE ONE FOR INDEX 14083000
ELSE 14084000
IF BOOARRAYID < TYPEV < INTARRAYID THEN 14085000
14086000

```



```

BEGIN 14087000
  GT1:=N:=TAKE(BUP+T1:=G,INCR); 14088000
  G,ADDRESS:=F:=F+N:=N+1; %LEAVE ROOM FOR LOWER BOUND 14089000
  WHILE N:=N-1#0 DO 14090000
    IF T2:=TAKE(BUP+T1+N) < 0 THEN 14091000
      PUT(ABS(T2&(4*(F=N)+2)LPB),BUP+T1+N) 14092000
    END 14093000
  ELSE G,ADDRESS:=F:=F+1; 14094000
  PUT(G,BUP); G,INCR:=GT1; % UPDATE PARAMETER INFO 14095000
  IF FWDTOG THEN 14096000
    BEGIN 14097000
      IF (GT1:=TAKE(MARK+PJ)),CLASS#G,CLASS THEN 14098000
        FLAG(49); % CLASS ERROR FORWARD TO ACTUAL 14099000
        IF GT1.VO#G.VO THEN FLAG(50); 14100000
      END 14101000
    ELSE 14102000
      PUT(G,MARK+PJ); % PROCEDURE ADDITIONAL INFO ENTRY 14103000
    END 14104000
  UNTIL PJI=PJ=1=0; % 14105000
  SPEBTOGI=FALSE; 14106000
  IF IPCJOB THEN % WRAP UP IPC SET UP & START COMPILE 14107000
    BEGIN 14108000
      IPCJOB:=BOOLEAN(2*MARK); % FALSE = POINTS TO + INFO 14113000
      IF STEP1#BEGINV THEN FLAG(55); 14114000
      ELBAT[1],CLASS:=SEMICOLON; 14115000
      GO TO START; % NOW WE CAN COMPILE THE IPC JOB 14116000
    END; 14117000
    GO TO HF; % NOT IPC, GO COMPILE THE PROCEDURE 14118000
  END SPECIFICATION WRAP UP; 14119000
  STOPDEFINE:=DUMPV*(GTA1[J+1]:=ELBAT[1],ADDRESS)#MONITORV; 14129000
  ERRORTOGI=TRUE; 14130000
  IF IPCJOB THEN % 14133100
  IF SPECTOG THEN % VERIFY THAT PROPER TYPES ARE BEING PASSED 14133200
    IF BOOV <= G:=GTA1[J] <= INTV OR G=ARRAYV OR G=EVENTV THEN % OK 14133300
      ELSE FLAG(550); % 14133400
  AUXSIZERR; % 14135500
  TASKDEC; TASKE; GO TO GOTSCHK; 14155500
  EVENTDEC; P3:=TRUE; ENTER(EVENTID); GO START; 14155700
  SCATTERELBAT; 14102100
  IF IPCJOB THEN % MAKE "MYSELF" TASK ARRAY ENTRY AT R+@25 14102200
    BEGIN % 14302300
      TYPEV:=TASKID; COUNT:=6; SCRAMT:=(ACCUM[1]#"6MYSEL") MOD 125; 14302400
      ACCUM[2]:=0#"F"[1:43:5]; 14302500
      GO DOITANYWAY; 14102600
    END; % 14302700
  $ VOID 14305001 14305000
  IF NOT IPCJOB THEN 14323000
  IF LEVEL < 91 THEN LEVMODE[LEVEL:=LEVEL+1]:=MODE ELSE FLAG(39); 14323100
  PJI=0; 14323200
  $ VOID 14324001 14324000
  IF STEP1#SECRET THEN FLAG(010) % WAS NOT IN PARAMETER LIST 14345000
  ELSE 14346000
    BEGIN 14347000
      G:=ELBAT[1]; 14348000
      IF 1<(IF IPCJOB THEN G,PJFLD ELSE G,ADDRESS)<=PJ THEN 14349000
        ELSE FLAG(10); % WAS NOT IN FORMAL PARAMETER LIST 14350000
      PUT(TAKE(G)&1[10:47:1],G); % MARK AS VALUE PARAMETER 14351000
    END % 14352000

```

\$ VOID 14354001	14353000
IF IPCJOB AND PJ=0 THEN % 0 PARAM IPC PROCEDURE DEC	14177000
GO TO IPCCOMPLETE; %	14178000
IF SPECTOG THEN GO TO START; % AND PROCESS THE SPECIFICATIONS	14379000
IF STEPI=FORWARDV OR Q="EXTERNAL" % EXT ILLEGAL PARAM CK IN PROCESS	14467000
IF ECCLASS # FORWARDV THEN % EXTERNAL	14469025
BEGIN	14469050
IF PROINFO,CLASS # PROCID THEN FLAG(558) % MUST BE UNTYPE	14469075
ELSE PUT(TAKE(PROINFO)&IPCPROCID CLASS,PROINFO);	14469100
END %	14469125
ELSE %	14469150
\$# PATCH NUMBER 411 FOR XALGOL CONTAINS 22 CARDS	
\$! THIS DOCUMENTS THE IPC ERROR MESSAGES, 1/5/73	
021 BLOCK; SAVE OR AUXMEM APPEARS IMMEDIATELY	00041000
BEFORE IDENTIFIER (NO TYPE),	00042000
550 IPCPROCEDUREDEC; AN IDENTIFIER OF THIS CLASS CAN NOT BE	00106900
PASSED TO AN INVOKED PROGRAM.	00306905
551 IPCPROCEDUREDEC; MISSING BEGIN AFTER SPECIFICATION PART,	00106910
552 ACTUALPARAPART; THIS INTRINSIC CAN NOT BE PASSED AS A	00106915
FORMAL PARAMETER.	00306920
553 PROCESSTATEMENT; A TASK IDENTIFIER MAY NOT BE PASSED AS AN	00306925
EXPLICIT PARAMETER TO AN INVOKED PROCEDURE,	00306930
554 BIMPFUN; PARAMETER MUST BE AN EVENT,	00306935
555 BIMPFUN; AN INVOKED PROCESS CAN NOT FIX OR FREE AN	00306940
EVENT PASSED TO IT.	00306945
556 BIMPFUN; MISSING COMMA IN ACQUIRE OR PEND FUNCTION,	00306950
557 IPCSTATEMENT; PARAMETER MUST BE AN EVENT,	00106955
558 BLOCK; AN EXTERNAL IPC PROCEDURE MUST BE UNTYPED,	00306960
559 PROCESSTATEMENT; PROCEDURE MUST BE DECLARED EXTERNAL,	00306965
560 PROCESSTATEMENT; MISSING OR INCORRECT TASK PART,	00306970
561 TASKATTRIBUTEHANDLER; MISSING , IN TASK ATTRIBUTE PART,	00306975
562 TASKATTRIBUTEHANDLER; MISSING OR UNDEFINED TASK ATTRIBUTE,	00306980
563 TASKATTRIBUTEHANDLER; TASK ATTRIBUTE IS NON ASSIGNABLE,	00306985
564 TASKATTRIBUTEHANDLER; MISSING ASSIGNMENT OPERATOR IN TASK	00306990
ATTRIBUTE ASSIGNMENT STATEMENT,	00306995
\$# PATCH NUMBER 412 FOR XALGOL CONTAINS 8 CARDS	
\$! THIS PROVIDES THE "RUN" STATEMENT FOR IPC, ONLY VALUE PARAMETERS	
\$! CAN BE PASSED TO PROCESSES THAT IS "RUN", 3/19/73,	
565 PROCESSTATEMENT; PARAMETERS TO A "RUN" TASK MUST BE VALUE,	00307000
% AND ! RUN !PCVALUEONLYPROCEDURE(AP1,AP2)[TASKID];	07765035
REAL E,T; BOOLEAN RUNNER; % SET TRUE IF CALLED AS A "RUN"	07565050
RUNNER:=Q,[42:6]=0; % 5TH CHARACTER OF "RUN" = 0, PROCESS = E,	07565135
IF RUNNER THEN FAULT(565); % ALL RUN PARAMS MUST BE VALUE	07565435
IF RUNNER THEN EMIT(55N); % STARTUSERTASK USES = FOR A "RUN"	07565815
IF Q="7PROCE" OR Q="3RUN00" THEN % IPC INITIATION	07810245
BEGIN PROCESSTATEMENT; GO XXX END; %	07810250
\$# PATCH NUMBER 413 FOR XALGOL CONTAINS 2 CARDS	
\$! MAKES TEMPORARY ASSIGNMENT FOR EXTERNAL PROCEDURE DECLARATIONS,	
\$! 3/20/73	
SPRTI((PRTIMAXI=PRTIMAX*1),[38:5]); % REUSE PRT CELL	14469110
BOOLEAN(*) AND NOT BOOLEAN(2*PRTIMAX,[43:5]);	14469115
\$# PATCH NUMBER 414 FOR XALGOL CONTAINS 1 CARD	
\$! STOPS DEFINES WHEN PICKING UP ATTRIBUTE NAME, 3/22/73	
STOPDEFINE:=TRUE; STEPIT; % PICK UP ATTRIBUTE NAME * NO DEFINES	07566425
\$# PATCH NUMBER 415 FOR XALGOL CONTAINS 6 CARDS	
\$! THIS CHANGE INSURES THAT THE LEXICOGRAPHICAL LEVEL OF THE TASK ID	00000010
\$! IN A PROCESS STATEMENT IS NO LOWER THAN THE "DEEPEST" NAME	00000020

```

S1 PARAMETER IN AN INVOKATION, 8/4/73 DHB 00000030
566 PROCESSTATEMENT; LEXICAL LEVEL OF TASK LESS THAN A NAME PARAM, 00307005
    INTEGER N, MARK, PCNT, MAXLVL; % 07765075
    IF STEPI # T, CLASS THEN FAULT(123); % WRONG TYPE 07565450
    IF (E1=ELBAT(I)), LVL > MAXLVL THEN MAXLVL=E, LVL; % 07565465
    IF (E1=ELBAT(I)), LVL < MAXLVL AND MAXLVL > 0 THEN FAULT(556); % 07565835
    EMITPAIR(E, LOD); % 07565850
S# PATCH NUMBER 416 FOR XALGOL CONTAINS 4 CARDS
S1 THIS CHANGE INSURES THAT AN IPC PROCEDURE IS NOT COMPILED TO "GO" AND 00000010
S1 CHECKS TO SEE THAT NO MORE THAN 29 PARAMETERS ARE ALLOWED IN AN 00800020
S1 IPC PROCEDURE DECLARATION, (SEE MCP PATCH 708) 8/4/73 DHB 00800030
567 PROGRAM; COMPILE AND GO OF INVOKED IPC PROCEDURE NOT ALLOWED, 00307010
568 BLOCK; AN INVOKED IPC PROCEDURE CAN HAVE ONLY 29 PARAMETERS, 00307015
    IF SAVETIME=0 THEN BEGIN ERRORLOG:=TRUE; FLAG(567) END; %GO WHERE 09252250
    IF IPCJOB THEN IF PJ > 29 THEN FLAG(568); % TOO MANY PARAMETERS 14335000
S# PATCH NUMBER 417 FOR XALGOL CONTAINS 5 CARDS
S1 THIS ADDS A NEW TASK ATTRIBUTE "INITIATOR" WHICH CAN BE SET AND READ, 00000010
S1 IF "INITIATOR" IS NON ZERO WHEN A TASK IS INITIATED IT WILL APPEAR AS 00000020
S1 IF IT ORIGINATED FROM THE STATION WITH THE INDICATED INTERNAL LINE 00000030
S1 NUMBER, 11/21/73 DHB 00000040
    IF Q = "7STACK" THEN 4 ELSE 07566575
    IF Q = "9INITI" THEN 5 ELSE 07566580
    562 * 07766599
    = 562 * 07766600
    EMITL(6); EMITL(1); EMITL(GTA1[0]); % SIZE, # DIMISIONS, # TASK 13595325
S# PATCH NUMBER 418 FOR XALGOL CONTAINS 2 CARDS
S1 THIS PROVIDES AN INDEXED ARRAY ROW WRITE OF THE FORM; 00000010
S1 WRITE(FIL, AEXP, ARRY[INDX]); 00000020
S1 IT IS THE USERS RESPONSIBILITY TO INSURE THAT AEXP+INDX DOES NOT 00000030
S1 EXCEED THE ARRAY BOUNDS, 11/24/72 00000040
    , "410" % LOCAL PATCH LEVEL 11/24/72 01831100
S VOID 08672001 08668000
S# PATCH NUMBER 419 FOR XALGOL CONTAINS 5 CARDS
S1 THIS PROVIDES THE CAPABILITY TO PROGRAMATICALLY GENERATE A MEMORY
S1 DUMP IN ALGOL OR XALGOL BY THE STATEMENT; 1/15/73
S1 DUMPNOW(INTEGER);
    , "411" % LOCAL PATCH LEVEL 1/15/73 DHB 01831100
    IF Q="7DUMPN" THEN % DUMPNOW(W) MEMORY DUMP 07810500
    BEGIN PANA; EMITL (0); EMITPAIR(48, COM); % 07810525
    EMIT0(DEL); EMIT0(DEL); GO TO XXX; % 07810550
    END; % 07810575
S# PATCH NUMBER 420 FOR XALGOL, XV, 1 CONTAINS 62 CARDS
S1 THIS MAKES EFFICIENCY IMPROVEMENTS TO THE XREF AND BEND PATCH 7/29/73 00800010
S VOIDT 01001776 01001750
ARRAY XREFAY1[0:10], XREFAY2[0:29], XINFO[0:31, 0:127]; % FOR XREF ONLY 01002000
INTEGER XREFPT, XLUN; % 01002100
DEFINE XREFINFO[R, C]=XINFO[R, (C) DIV 2]; % 01002200
    XMARK=IF XREF THEN XREFAY2[XREFPT-1]=NABS(*) % MARK ASSIGNMENT 01002300
ARRAY BEGINSTACK[0:31]; INTEGER BSPPOINT; % FOR BEND 01002400
BOOLEAN DEFINING; % BEND 01002500
S VOIDT 01007651 01007100
    STREAM PROCEDURE STUFF(LVL, N, L); VALUE LVL, N; 02196530
    DI=L; DS=8 LIT " "; SI=L; DS=13 WDS; DI=DI-4; 02196550
    DS=LIT[" "; SI=LOC LVL; DS=2 DEC; DS=LIT[" "; DS=8 DEC; 02196560
    STUFF(BSPPOINT, BEGINSTACK[ BSPPOINT ], LIN); 02196580
    IF Q = "4XREF0" THEN BEGIN SWITCHIT(XREFBIT); GO AGAIN END; 02419100
S VOID 02419121 02419110

```

Not in ucsc

```

IF NOT ENDTOG THEN % DONT XREF COMMENTS FOLLOWING END          02882050
IF T,[95:13] > SORTA THEN % NOT "ALGOL" IDENTIFIER           02882200
PTOG:=TRUE; % DONT XREF THESE THINGS                          05442500
PTOG:=FALSE; %                                               05443500
      IF XREF THEN % DONT COUNT THIS OCCURANCE IF10264802
      XREFPT=XREFPT-REAL(ELBAT[1],LVL#0);%SEEN B410264804
IF XREF THEN % DONT XREF SECOND DEC OF SAME VAR                13348000
  XREFPT=XREFPT-REAL(LEVEL#0); % GET RID OF LAST XREF13348100
  END; %                                                        13348200
  DEFINING:=TRUE; % WE ARE                                     14255500
  IF XREF THEN % DONT COUNT THIS AS AN OCCURANCE              14259077
  XREFPT=XREFPT-REAL(ELBAT[1],LVL#0); % IF SEEN B4            14259078
  PTOG:=TRUE; E; PTOG:=FALSE; % DONT XREF DEFINE PARAMS14259096
  DEFINING:=FALSE; % WE ARE NOT                               14268500
  IF XREF THEN % FORGET THE PARAMETERS SPECIFED IN A FORWARD  14470900
  IF G:=TAKE(MARK) > 0 THEN % OR EXTERNAL DECLARATION        14470910
  BEGIN SPACE(DSKI,"G"); XLUNT=XLUN=G END;                     14470920
IF XLUN > 0 AND ERRORCOUNT = 0 THEN % WE DID SOME XREFFING SO DOITTOIT 17001000
  IF NOT NOHEADING THEN WRITE(LINE[PAGE]); DATIME;            17002520
$ VOIDT 1705555R                                             17047001
  STREAM PROCEDURE PUP(S,D);                                   17048000
  BEGIN                                                       17049000
    ST:=S; DT:=D;                                             17050000
    60(IF SC=" " THEN JUMP OUT; DS:=CHR);                       17051000
    DS:=10LIT" "; S1:=S; S1:=S1+32; S1:=S1+28; % TO SEG NUM  17052000
    DST=4CHR; DS:=3LIT" *"; DS:=8CHR; DS:=LIT"*";            17053000
  END;                                                         17054000
  DEFINE BLANK(P) = BLANKET(14,P)#,                             17063000
  PAY=LIN#;                                                    17064000
$ VOID 17069999                                             17065000
  DO BEGIN                                                    17073500
    IF XREFPT=XREFPT+1=30 THEN                                  17074000
    BEGIN                                                      17075000
      READ(DSK2,30,XREFAY2[*])[EOF];                          17076000
      XREFPT=0;                                                17077000
    END;                                                       17078000
    IF (I=XREFAY2[XREFPT]),[21:27]>99999999 THEN GO TO EOF; 17079000
  END;                                                         17079250
  UNTIL I,[9:12] GTR 0; % DONT SORT NON XREFED ENTRIES      17079500
  BLANK(PAY);                                                  17083000
  IF LASTADDRESS NEQ LASTADDRESS:=A[0] THEN % NEW OCCURANCE  17094100
  IF LASTADDRESS,[9:12] GTR XREFAY1[9] THEN                  17095100
  BLANK(PAY); XREFPT:=0;                                       17099000
$ VOID 17100101                                             17100000
  BLANK(PAY);                                                 17101000
  PUP2(LASTADDRESS LSS 0,LASTADDRESS,[21:27],XREFPT,PAY[1]); 17103000
$ VOID                                                       17103010
  BLANK(PAY);                                                 17103500
$# PATCH NUMBER 421 FOR XALGOL CONTAINS 287 CARDS           00000020
$! THIS IMPLEMENTS THE B6700 "INCLUDE" $ CARD OPTION AND DE= 00000010
$! IMPLEMENTS THE "CAST" LIBRARY COMPILER FEATURE, IN ADDITION THIS MAKES SEVERAL 00000020
$! IMPROVEMENTS TO THE READACARD ROUTINE, THE HANDELING OF VOID CARDS 00000030
$! IS CORRECTED AND THE "EOF NO LABEL" PROBLEM IN THE COMPILER IS 00000040
$! ELIMINATED, THIS LAST CHANGE MAKES 9#S CARDS OPTIONAL ON BOTH THE 00000050
$! CARD AND TAPE FILES,                                       00800060
$! NOVEMBER 20, 1973. -- DAVE BROWN -- U.S. CUSTOMS SAN DIEGO, CA. 00000070
$ VOIDT 0030508;                                             00305010

```

*Superseded by / conflicts with
official patch 108*

615	READACARD; UNEXPECTED END OF INPUT.	00415000
620	STARTINCLUDING; NESTED INCLUDES NOT IMPLEMENTED,	00420000
621	STARTINCLUDING; FIRST INCLUDE CARD MUST NAME A FILE,	00421000
622	STARTINCLUDING; ILLEGAL FILE IDENTIFIER,	00422000
623	STARTINCLUDING; ILLEGAL SEQUENCE NUMBER ON INCLUDE CARD,	00423000
624	STARTINCLUDING; FILE MUST HAVE BEEN USED BEFORE FOR * OPTION,	00424000
625	STARTINCLUDING; BAD SYNTAX ON THIS INCLUDE CARD,	00425000
626	STARTINCLUDING; YOUR INCLUDE FILE IS NOT ON DISK,	00426000
627	STARTINCLUDING; BEGINNING SEQUENCE NUMBER IS NOT IN THE FILE,	00427000
	INTEGER ADDVALUE, TOTALNO; %	01000860
	\$ VOIDT 01001521	01001490
	\$ VOIDT	01331000
	THE LAST CHARACTER IN THE TAPE AND CARD BUFFERS, ;	01336000
	\$ VOIDT 01339001	01337000
	0 INCLUDE IN PROCESS,	01394700
	> 3 ELABORATING A DEFINE.	01398000
	\$ VOIDT 01398201	01398100
	SAVE ARRAY DEFINEARRAY[0:25]; % 013*(MAX NEST DEPTH)+1 (SEE 1498100)	01491000
	DEFINE LASTSEQ = DEFINEARRAY[25]; % TEMP STORAGE FOR SEQ OF CARD	01498100
	LSEQM1 = DEFINEARRAY[24]; % FOR CALLING CONV	01498200
	LASTERR = DEFINEARRAY[24]; % HOLDS SEQ NO OF LAST ERROR	01498300
	FILE INCLUDE DISK SERIAL (1,10,150);	01561010
	\$ VOIDT 01561051	01561011
	FILE DSK1 DISK SERIAL [20:300] (2,10,30); % FOR XREF	01561025
	FILE DSK2 DISK SERIAL [20:300] (2,30,30); % FOR XREF	01561050
	SAVE ARRAY CBUFF, TBUFF, IBUFF[0:9]; % INPUT BUFFERS	01561056
	\$ VOIDT 01561451	01561065
	\$ VOIDT 01576001	01575000
	\$ VOIDT 01743301	01742200
	\$ VOIDT 01784001	01756000
	INTEGER SAVLASTUSED, ILCR; % FOR INCLUDE	01783000
	REAL INCLCR, INCLRANGE; % FOR INCLUDE OPTION	01784000
	, " = 413" % LOCAL PATCH LEVEL 11/20/73 DHB	01831100
	\$ VOIDT 02001611	02001010
	\$ VOIDT 02013721	02013010
	BOOLEAN PROCEDURE STARTINCLUDING; FORWARD; %	02065700
	BOOLEAN PROCEDURE READINCLUDE; FORWARD; %	02065800
	\$ VOIDT 02134501	02131500
	PROCEDURE WRITEBINE; %	02181000
	IF SINGLTOG THEN WRITE(LINE, 15, LIN[*]) %	02181250
	ELSE WRITE(LINE[DBL], 15, LIN[*]); %	02181500
	\$ VOIDT 02182251	02181750
	DEFINE PRINTCARD = BEGIN %	02182500
	DSI=LIT" "; SII=LOC SYMBOL; SIII=SII+7;	02185250
	DSI=CHR; DSI=2LIT" ";	02185500
	SWITCH SWI=LICARD, LCARD, LTAPE; %	02191250
	CARDNUMBER:=IF TOTALNO < 0 THEN TOTALNO:=ABS(TOTALNO)-1	02192000
	ELSE TOTALNO:=TOTALNO+ADDVALUE;	02192250
	\$ VOIDT 02193001	02192500
	CHANGESEQ(CARDNUMBER, LCR);	02193250
	IF LASTUSED > 0 OR NEWINCL THEN	02193800
	IF SAVLASTUSED=3 THEN GO TO LTAPE; % INCLUDE CARD WAS ON TAPE	02194600
	\$ VOIDT 02202251	02198500
	DEFINE SEQCOMPARE(TLCR, CLCR) = %	02202500
	\$ VOIDT 02203001	02202750
	MEDIUM="C"; % FROM CARD READER	02203250
	LORI=TLCR; LASTUSED:=3;	02204000

```

MEDIUM="T" % FROM TAPE
MEDIUM="P" % CARD PATCHES TAPE
LASTUSED=3 % IN CASE OF EOF ON TAPE
READ(TAPE,10,TBUFF[*])[EOF]

$ VOIDT 02208501
    LASTUSED=2;
    #, % END SEQCOMPARE DEFINE
    DIDINCLUDE=REAL(DOLLARTOG)="H"#; %
    LABEL USETHESWITCH,CARDONLY,CARDLAST,EOF,TAPELAST,COMPAR,EXIT,XIT;
    BOOLEAN DOLLAR2TOG; %
$ VOIDT 02211251
    IF LASTUSED=1 THEN GO TO CARDONLY; %
    IF LASTUSED=3 THEN GO TO TAPELAST; %
    IF LASTUSED=2 THEN GO TO CARDLAST; %
    IF LASTUSED=0 THEN % INCLUDE
    IF READINCLUDE THEN GO USETHESWITCH ELSE GO EXIT; %
$ VOIDT 02215001
    LCR=CLCR; %
$ VOIDT
    READ(CARD,10,CBUFF[*])[EOF];
    IF LASTUSED=1 THEN GO TO EXIT; % CARD ONLY
    IF LASTUSED=1 THEN % EOF ON CARD MEANS BYE=BYE
        BEGIN ERRORTOG=TRUE; ERR(615) END; % BRANCHES TO ENDOFITALL
    IF LASTUSED=3 THEN % EOF ON TAPE = SWITCH TO CARD ONLY
        BEGIN LCR=CLCR; LASTUSED=1; MEDIUM="C"; GO EXIT END;
    FILL CBUFF[*] WITH 4("% END=OF","=FILE *#"),"COMPILER","99999999";
$ VOIDT 02218401
$ VOIDT 02219001
    READ(TAPE,10,TBUFF[*])[EOF];
$ VOIDT 02224001
    SEQCOMPARE(TLCR,CLCR);
$ VOIDT
$ VOIDT
    IF COMPARE(MKABS(LASTSEQ),LCR)=1 THEN %
    PUTSEQNO(LASTSEQ,LCR); %
    CARDNUMBER=CONV(LSEQM1,5,8); %
    NCR=NCR+32768; % OVER "$" (CHAR MODE POINTER)
    DOLLARCARD; %
    IF NOT DIDINCLUDE THEN %
    IF LISTPTOG OR PRINTDOLLARTOG THEN PRINTCARD;
    IF DOLLAR2TOG=EXAMIN(FCR)=" " AND EXAMIN(FCR+32768)="$" THEN
$ VOIDT
$ VOIDT
    NCR=NCR+32768+32768; % OVER "$" (CHAR MODE POINTER)
    IF DIDINCLUDE THEN GO USETHESWITCH; %
$ VOIDT 02235251
    OUTPUTSOURCE;
    CARDCOUNT=CARDCOUNT+1;
    IF OMITTING OR %
    DOLLAR2TOG THEN GO USETHESWITCH;
$ VOIDT
    OPTIONS[OPINX+1]=REAL(BI=TRUE); %
$ VOIDT
$ VOIDT
$ VOIDT
ALONGI
$ VOIDT

```

```

02204250
02205500
02205750
02206000
02206250
02209250
02209750
02210000
02210250
02210500
02210750
02212000
02212025
02212050
02212100
02212200
02213500
02215500
02215750
02216250
02216500
02217250
02217275
02217500
02217750
02218000
02218250
02218750
02219500
02219750
02224500
02224750
02225500
02226000
02228000
02228100
02228750
02229000
02229200
02229250
02230000
02230100
02230500
02230750
02231100
02234251
02234500
02234750
02235000
02235250
02235500
02334000
02335000
02339000
02353000
02356500
02358000

```

```

STREAM PROCEDURE GETVOID(VP,NCR,VR,LCR,SEQ); VALUE NCR,LCR;
  BEGIN
    LABEL L,TRANS;
    SI:=NCR; DI:=VP; DS:=8 LIT "0";
    2(34(IF SC="" THEN SI:=SI+1 ELSE JUMP OUT 2 TO L));
    SI:=SEQ; NCR:=SI; TALLY:=8; GO TRANS; % USE SEQ NO OF THIS CARD
  L;
    IF SC="" THEN
      BEGIN
        SI:=SI+1; DI:=LCR; DS:=1 LIT""; % STOPPER FOR SCAN
        NCR:=SI; % TEMP, STORAGE, SINCE NCR IS "LOCAL" TO GETVOID,
        8(IF SC="" THEN JUMP OUT ELSE
          BEGIN TALLY:=TALLY+1; SI:=SI+1 END);
        END
      ELSE BEGIN
        NCR:=SI; % TEMP, STORAGE, SINCE NCR IS "LOCAL" TO GETVOID,
        DI:=LCR; DS:=1 LIT""; % STOPPER FOR SCAN
        8(IF SC="" THEN JUMP OUT ELSE
          BEGIN TALLY:=TALLY+1; SI:=SI+1 END);
        END;
      TRANS;
        SI:=NCR; DI:=VP; DI:=DI+8; % RESTORE POINTERS,
        NCR:=TALLY; DI:=DI-NCR; DS:=NCR CHR;
        DI:=DI+8; VP:=DI; % I.E. "LOC VP"=DI
        DI:=VR; SI:=LOC VP; DS:=WDS; % ADDRESS OF VOID RANGE,
        END OF GETVOID;
        DOLLARTOG:=DOLLARTOG OR TRUE; %
      $ VOIDT
        GETVOID(VOIDPLACE,NCR,VOIDCR,LCR,LASTSEQ); %
      $ VOIDT
        IF TLCR = 0 THEN % FIRST TIME ONLY
          TLCR:=MKABS(TBUFF[9]);
        $ VOIDT
          GETVOID(VOIDPLACE,NCR,VOIDCR,LCR,LASTSEQ); %
        $ VOIDT
          IF Q = "7INCLU" THEN %
            BEGIN DOLLARTOG:=STARTINCLUDING; GO EXIT END;
          IF Q = "7INCLN" THEN %
            BEGIN SWITCHIT(NEWINCLBIT); GO AGAIN END; %
            TOTALNO:=-(CONV(ACCUM[1],0,COUNT)+1);
          $ VOIDT 0258300;
            CHANGESEQ(CARDNUMBER,LCR); % RESTORE SEQ FIELD FOR VOIDRANGE CHECKS
            DOLLARTOG:=DOLLARTOG AND NOT TRUE; %
            CROSSHATCH; %
            RTPAREN,CROSSHATCH; %
          DOLLARCARD;
          $ VOIDT 0273600;
          $ VOIDT 0290800;
          $ VOIDT 02999999;
          BOOLEAN PROCEDURE READINCLUDE;
          BEGIN
            LABEL EOF; %
            LCR:=ILCR; MEDIUM:=""");
            READ(INCLUDE,IO,IBUFF[*])(EOF);
            IF COMPARE(ILCR,INLCR)=1 THEN % THIS CARD OUT OF RANGE
          EOF;BEGIN
            LASTUSED:=SAVLASTUSED; MEDIUM:="C"; % IN CASE OF CARD ONLY
            READINCLUDE:=TRUE;

```

```

02359100
02359125
02359150
02359175
02359200
02359225
02359250
02359275
02359300
02359325
02359350
02359375
02359400
02359425
02359450
02359475
02359500
02359525
02359550
02359575
02359600
02359625
02359650
02359675
02359700
02359725
02366000
02368000
02410000
02411000
02439000
02456000
02458000
02485000
02486000
02570000
02571000
02572000
02573000
02581000
02582000
02602800
02603000
02640000
02642000
02530000
02732000
02907000
02985001
02986000
02986100
02986200
02986300
02986400
02986500
02986600
02986700
02986800

```

```

        BLANKET(0, LASTSEQ); %
        CLOSE(INCLUDE);
    END;
END READINCLUDE;
BOOLEAN PROCEDURE STARTINCLUDING;
BEGIN
    LABEL AWAY, THRU;
    REAL FIRST, T; INTEGER I, J, K;
    BOOLEAN SVNEWTOG;
    DEFINE FLAGIT(E)=BEGIN FLAG(E); GO AWAY END#;
        FINDFIRST=SVNEWTOG#; MAXR=T#; NEW=FIRST#;
    IF OMITTING THEN GO AWAY;
    IF VOIDING OR VOIDTAPE THEN
        IF COMPARE(MKABS(LASTSEQ), VOIDCR)=0 THEN %
            IF VOIDTAPE AND LASTUSED=3 OR NOT VOIDTAPE THEN GO AWAY;
    CHANGESEQ(CARDNUMBER, LCR); % RESTORE FOR LIST
    SVNEWTOG=NEWTOG;
    NEWTOG=NEWTOG AND NOT NEWINCL;
    OUTPUTSOURCE; CARDCOUNT:=CARDCOUNT+1; %
    NEWTOG=SVNEWTOG;
    TURNONSTOPLIGHT("%", LCR); % FOR SCANNING
    STARTINCLUDING:=BOOLEAN("H"); % TELL READACARD WE DID OUR THING
    IF LASTUSED=0 THEN FLAGIT(620); % CANT NEST THEM
    SKAN;
    IF SAVLASTUSED=0 AND RESULT#1 THEN FLAGIT(621); % MUST NAME IT FIRST
    IF RESULT=1 THEN % FILE ID
        BEGIN
            IF COUNT > 7 THEN FLAGIT(622);
            T:=" "; MOVECHARACTERS(COUNT, ACCUM[1], 3, T, 1);
            IF INCLUDE, MFID#T THEN INCLUDE, MFID:=NEW:=T;
            SKAN; IF RESULT=2 THEN SKAN;
            IF RESULT#1 OR COUNT > 7 THEN FLAGIT(622);
            T:=" "; MOVECHARACTERS(COUNT, ACCUM[1], 3, T, 1);
            IF INCLUDE, FID#T THEN INCLUDE, FID:=NEW:=T;
            SKAN;
        END;
    INCLRANGE:="99999999"; FINDFIRST:=FALSE;
    IF RESULT=3 THEN % NUMBER IS STARTING NUMBER
        BEGIN
            FINDFIRST:=TRUE; FIRST:=0;
            IF COUNT > 8 THEN FLAGIT(623);
            MOVECHARACTERS(COUNT, ACCUM[1], 3, FIRST, 8-COUNT);
            THRU; SKAN;
            IF Q="1*0000" OR Q="2*0000" THEN
                BEGIN % THRU SPECIFIER
                    SKAN;
                    IF RESULT#3 OR COUNT > 8 THEN FLAGIT(623); % BAD NUMBER
                    INCLRANGE:=0;
                    MOVECHARACTERS(COUNT, ACCUM[1], 3, INCLRANGE, 8-COUNT);
                    SKAN;
                END;
            END;
        END
    ELSE
        IF Q="1*0000" THEN % START WHERE WE LEFT OFF
            BEGIN
                IF SAVLASTUSED=0 OR NEW#0 THEN FLAGIT(624);
                MOVE(1, IBUFF[9], FIRST); FINDFIRST:=TRUE;

```

```

02986900
02987000
02987100
02987200
02987300
02987400
02987500
02987600
02987700
02987800
02987900
02988000
02988100
02988200
02988300
02988400
02988500
02988600
02988700
02988800
02988900
02989000
02989100
02989200
02989300
02989400
02989500
02989600
02989700
02989800
02989900
02990000
02990100
02990200
02990300
02990400
02990500
02990600
02990700
02990800
02990900
02991000
02991100
02991200
02991300
02991400
02991500
02991600
02991700
02991800
02991900
02992000
02992100
02992200
02992300
02992400
02992500

```


14121
Moore Business Forms, Inc. NY

```
GO TO THRU; % AND GET = SEQ NO
END;
IF Q#*1%0000* OR XMODE#0 THEN FLAGIT(625); % BAD SYNTAX
ILCRI=MKABS(IBUFF[9]);
SEARCH(INCLUDE,IBUFF[*]);
IF MAXR=IBUFF[5]+1 < 0 THEN FLAGIT(626); % FILE NOT ON DISK
IF FINDFIRST THEN % BINARY SEARCH FOR FIRST RECORD
BEGIN
  INLCRI=MKABS(FIRST);
  K:=MAXR+J1=1;
  WHILE J < K DO J1:=J+J; I:=J DIV 2;
  FOR KI=I STEP 1 IF J#0 THEN I ELSE =I
  WHILE I > 0 DO
  BEGIN
    IF K >= MAXR THEN J1=1 ELSE
    BEGIN
      READ(INCLUDE[K],10,IBUFF[*]);
      J1=COMPARE(ILCR,INLCR);
    END;
    I:=IF J#2 THEN 0 ELSE I DIV 2;
  END;
  IF J#2 THEN BEGIN CLOSE(INCLUDE); FLAGIT(627) END; % NOT THERE
END;
SAV-LASTUSED:=LASTUSED; LASTUSED:=0; % WE ARE INCLUDING
INLCRI=MKABS(INCLRANGE);
BLANKET(0, LASTSEQ); %
AWAY;
END START INCLUDING;
%
COMMENT#####
FORWARD DECLARATIONS
#####;
MOVE(1, LASTSEQ, LIN[12]);
WRITERROR(REMOTOG, ERRNUM, ACCUM[1], LIN, Q, [12:6], LASTSEQ);
% VOIDT
MOVE(1, LASTSEQ, PNCH[9]); %
IF ERRNUM=199 OR ERRNUM=200 OR %
ERRNUM=611 OR ERRNUM=615 THEN GO TO ENDOFITALL;
LASTUSED:=1; % CARD ONLY
TOTALNO:=(1+ADDVALUE:=1000);
% VOIDT
NEXTINFO:=LASTINFO+SORTA+1;
BLANKET(1, LASTERR); % BLANK LASTERR AND LASTSEQ
MEDIUM:="C"; LASTUSED:=1; % ASSUME CARD UNTIL TOLD DIFFERE
CCRI=MKABS(CBUFF[9]); %
READACARD; % READ FIRST CARD AND OPEN FILES
NEXTINFO:=LASTINFO+SORTA+1;
% VOIDT 0927555;
MOVECHARACTERS(4, LASTERR, 0, GT1, 4); %
MOVECHARACTERS(4, LASTERR, 4, GT2, 4); %
% VOIDT 0942600;
ERROR(199); % GOES TO ENDOFITALL
TIME:=TIME(1); % BOJ TIME
GETREADY; % READY SET --- GO ---
BLOCK(FALSE); %
ENDOFITALL;
CLEANUP; %
% PATCH NUMBER 422 FOR XALGOL CONTAINS 22 CARDS
```

02992600
02992700
02992800
02992900
02993000
02993100
02993200
02993300
02993400
02993500
02993600
02993700
02993800
02993900
02994000
02994200
02994300
02994400
02994500
02994600
02994700
02994800
02994900
02995000
02995100
02995200
02995300
02995400
03000000
03000100
03000200
03000300
05039500
05044000
05045000
05058000
05107100
05107200
07030000
09028920
09029000
09033000
09034500
09035000
09036000
09037000
09214980
09275100
09415000
09416000
09421000
13112500
17000000
17000100
17800200
17000300
17000400

```

$! THIS PROVIDES A BEGIN - END MATCH UP AS ON THE B6700, DMB 11/13/73 00000010
  INTEGER NEWBEGCT,OLDBEGCT) & TO MATCH BEGINS - ENDS 01002400
$ VOID 01002500
  EDITLINE(LIN,FCR,ABS(OLDBEGCT),L,[36:10],MEDIUM,OMITTING); 02182750
STREAM PROCEDURE EDITLINE(LINE,NCR,BEND,R,SYMBOL,OMIT); 02183500
  VALUE NCR,BEND,R,SYMBOL,OMIT; 02183750
  DS:=LIT" "; NCR:=DI; DI:=DI-17; 02185250
  SI:=LOC BEND; DS:=4 DECT DI:=DI-4; DS:=4 FILL; DI:=NCR; 02185300
  SI:=LOC SYMBOL; SI:=SI+7; DS:=CHR; DS:=2 LIT" "; 02185500
  OLDBEGCT:=0; & WE PRINTED IT 02196275
$ VOID 02196611 02196510
  IF DEFINECTR = 0 THEN & NOT IN A DEFINE DECLARATION 02869100
  IF T.CLASS = BEGINV THEN 02869200
  OLDBEGCT:=NEWBEGCT;NEWBEGCT+1 ELSE 02869300
  IF T.CLASS = ENDV THEN 02869400
  BEGIN 02869500
  OLDBEGCT:=IF OLDBEGCT > 0 THEN 0 ELSE =NEWBEGCT; 02869600
  NEWBEGCT:=NEWBEGCT-1; 02869700
  END; 02869800
$ VOIDT 02910801 02910100
  EDITLINE(LIN,FCR,0," ",MEDIUM,0); 05038000
$ VOIDT 14255500
$ VOIDT 14268500
$# PATCH NUMBER 423 FOR XALGOL CONTAINS 15 CARDS
$! THIS CORRECTS AND IMPROVES THE CODE GENERATED FOR CASE STATEMENTS, 00000010
$! 1. A CASE INDEX EQUAL TO THE NUMBER OF CASE ELEMENTS WILL NOW 00000020
$! CAUSE AN INVALID INDEX AS ALL OTHER INVALID CASE INDICIES DO, 00000030
$! 2. FOUR SYLLABLES OF UNNECESSARY BRANCHING CODE IS ELIMINATED AT 00000040
$! THE END OF THE CASE STATEMENT, 00000050
$! 3. AN EXTRA WORD IN THE CASE BRANCH SEGMENT IS REMOVED, 00000060
$! 1/8/74 00000070
  ,"-415" & LOCAL PATCH LEVEL 1/8/74 DMB 01831100
  REAL LINK, LLINK, TEMP, N, ADR, PRT, NULL; 07646390
  LABEL LOOP, XIT, ONOUT; 07646410
$ VOID 07646421 07646420
  END ELSE 07646500
  IF ELCLASS = ENDV THEN & AT ESAC 07646502
  BEGIN IF N=0 THEN N:=1; & IN CASE OF EMPTY CASE STATEMENT 07646504
  IF LINK > 0 AND NOT GOTOG THEN & OMIT LAST BRANCH IF ITS THERE 07646506
  BEGIN LINK:=LLINK; LI=L-2 END; 07646507
  GO ONOUT; 07646508
  END; 07646509
  BEGIN EMIT(LLINK:=LINK); LINK:=L:=L+1 END; 07646533
  END; & 07646538
$ VOID 07646551 07646540
ONOUT; 07646556
$# PATCH NUMBER 424 FOR XALGOL CONTAINS 107 CARDS
$! THIS IMPLEMENTS THE TRUTHSET AS ON THE B6700, IN ADDITION 00800010
$! THE DEFAULT TRUTHSET "NUMERIC" IS ADDED, 00000020
$! 1/13/74 00000030
  549 TRUTHSETI; TRUTHSET SPECIFIER MUST BE A STRING, 00306895
  TRUTHSETID #118#; COMMENT 166; 01278075
  EVENTV #23#; COMMENT 27; 01298700
  TRUTHSETV #24#; COMMENT 30; 01298750
  INTEGER NTSETS; & NUMBER OF TRUTHSETS DECLARED * 3 01519000
  ARRAY TRUTHSETS(0:119); & CAN HAVE 40 01520000
  POWERSOFTEN =(SORTA=1)#, 01574000

```

NOT IN UCSC

SORTA	=765#; % MARK XV.2,0 2/13/74	01580000
"=424"	% LOCAL PATCH LEVEL 1/13/74	01831100
STREAM PROCEDURE	DEBUGWORD(SEQ, CODE, FEIL); VALUE SEQ;	04130000
	SI=CODE;	04134000
	IF ACLASS=TRUTHSETID AND NOT VBIT THEN GO TO L22; % NO VALUE	07217513
	DEFINE CNA = 29#; % CHARACTER IN ALPHA	07957410
	CNN = 30#; % CHARACTER IN NUMERIC	07957420
	CNT = 31#; % CHARACTER IN TRUTHSETID	07957430
	EMIT(CMKS); EMITL(C); EMITL(C); EMITL(C);	07957440
\$ VOID	07957701	07957700
	EMITL(C);	07957750
	EMITL(CNA); % CHARACTER IN ALPHA	07957800
\$ VOID	07957901	07957900
	END ELSE	07958010
	IF ELCLASS=TRUTHSETID THEN	07958020
BEGIN		07958030
	SIMPLENAME(ELBAT(I)); EMITL(CNT); STEPIT;	07958040
END ELSE		07958050
	IF Q="7NUMER" THEN	07958060
BEGIN		07958070
	EMITL(C); EMITL(CNN); STEPIT;	07958080
	EMITV(GNAT(STRNGCOMPARE));	07958150
	SWN = 25#; % SCAN WHILE IN NUMERIC	07961811
	SWT = 26#; % SCAN WHILE IN TRUTHSETID	07961812
	SUA = 29#; % SCAN UNTIL IN ALPHA	07961820
	SUN = 30#; % SCAN UNTIL IN NUMERIC	07961821
	SUT = 31#; % SCAN UNTIL IN TRUTHSETID	07961822
	BEGIN EMITL(IF WHILECOND THEN SWA ELSE SUA); EMITL(C) END ELSE	07965400
	IF ELCLASS = TRUTHSETID THEN	07965410
	BEGIN EMITL(IF WHILECOND THEN SWT ELSE SUT);	07965420
	SIMPLENAME(ELBAT(I));	07965430
	END ELSE	07965440
	IF Q = "7NUMER" THEN	07965450
	BEGIN EMITL(IF WHILECOND THEN SWN ELSE SUN); EMITL(C) END ELSE	07965460
\$ VOID	07965601	07965600
	TWN = 25#; % TRANSFER WHILE IN NUMERIC	07968410
	TWT = 26#; % TRANSFER WHILE IN TRUTHSETID	07968420
	TUT = 29#; % TRANSFER UNTIL IN TRUTHSETID	07968430
	TUN = 30#; % TRANSFER UNTIL IN NUMERIC	07968440
	COMMENT CONDITION IS; WHILE IN ALPHA, NUMERIC, TRUTHSETID	07975200
	OR; UNTIL IN ALPHA, NUMERIC, TRUTHSETID;	07975300
	BEGIN EMITL(IF WHILECOND THEN TWA ELSE TUA); EMITL(C) END ELSE	07975500
	IF ELCLASS = TRUTHSETID THEN	07975510
	BEGIN EMITL(IF WHILECOND THEN TWT ELSE TUT);	07975520
	SIMPLENAME(ELBAT(I));	07975530
	END ELSE	07975540
	IF Q = "7NUMER" THEN	07975550
	BEGIN EMITL(IF WHILECOND THEN TWN ELSE TUN); EMITL(C) END ELSE	07975560
\$ VOID	07975801	07975800
	OCT0132000000560000, "8HAPPE", "NED00000", OCT2000000000100020, %74409214430	
	OCT04300000300000000, "8TRUTH", "SET00000",	%758 09214437
	OCT00300000240000000, "7PROTE", "CT000000",	%76109214440
	OCT20000000000004050, COMMENT POWERS OF TEN ;	%76409214444
	0; COMMENT SORTA ;	%76509214445
692 STEP 3 UNTIL	728, 732 STEP 4 UNTIL 748, 750 STEP 2 UNTIL 758,761	09214513
END;		09360100
NTSETS1=NTSETS-3;		09360200
FOR I1=0 STEP 3 UNTIL NTSETS DO		09360300

```

BEGIN 09360400
  MOVE(1,TRUTHSETS[1],PRT[(GT1:=TRUTHSETS[I+2]) DIV 128, 09360500
    GT1 MOD 128])) 09360600
  MOVE(1,TRUTHSETS[I+1],PRT[(GT1:=GT1+1) DIV 128, 09360700
    GT1 MOD 128])) 09360800
  IF BOOARRAYIDSTYPES<INTARRAYID 13330550
    OR TYPE = TASKID 13330560
    OR TYPE = TRUTHSETID 13330570
  THEN & MUST BE CALLED BY NAME 13330580
PROCEDURE TRUTHSETI; 13595500
BEGIN 13595510
  STREAM PROCEDURE TRUTHSETTER(N,A,T); VALUE N; 13595520
  BEGIN LOCAL C,CP; 13595530
    SI:=A; SII:=SI+3; DI:=LOC C; DII:=DI+7; CPI:=DI; 13595540
    N ( DII:=CPI; DS:=CHR; DII:=T; SKIP DB; SKIP C DB; DS:=SET ); 13595550
  END SETTER; 13595560
  LABEL AWAY; 13595570
  IF GTA1[J=1]#0 THEN FLAG(25); % NO SAVE OR OWN 13595580
  I:=I+1; 13595590
  DO BEGIN 13595600
    STEPIT; P2:=TRUE; ENTRY(TRUTHSETID); 13595610
    IF SPECTOG THEN GO AWAY; 13595620
    TRUTHSETS[NTSETS+2]:=GETSPACE(TRUE,=12)=1; % CONSTANT OR STRING; 13595630
    IF ELCLASS#LEFTPAREN THEN BEGIN ERR(105); GO AWAY END; 13595640
    IF STEPI#STRING AND ELCLASS#STRNGCON THEN % BAD 13595650
      BEGIN ERR(549); GO AWAY END; 13595660
    DO TRUTHSETTER(COUNT,ACCUM[1],TRUTHSETS[NTSETS]) 13595670
    UNTIL STEPI#STRING AND ELCLASS#STRNGCOM; 13595680
    IF ELCLASS#RTPAREN THEN BEGIN ERR(104); GO AWAY END; 13595690
    IF DEBUGTOG THEN 13595691
      BEGIN DEBUGWORD("  ",TRUTHSETS[NTSETS],LIN); WRITELINE; 13595692
      DEBUGWORD("  ",TRUTHSETS[NTSETS+1],LIN); WRITELINE; 13595693
    END DEBUG; 13595694
    NTSETS:=NTSETS+3; 13595700
  END 13595710
  UNTIL STEPI#COMMA; 13595720
AWAY; 13595730
END TRUTHSETDECER; 13595740
  TRUTHSETDEC, % 14016750
  PTRDEC,DEFINDEC,AUXSIZERR,AUXSIZERR,TASKDEC,EVENTDEC, 14021000
  TRUTHSETDEC; 14021250
TRUTHSETDEC: TRUTHSETI; GO TO START; 14155800
%# PATCH NUMBER 425 FOR XALGOL CONTAINS 4 CARDS 00000000
%# THIS ADDS THE NUMBER OF SYNTACTIC ITEMS PROCESSED TO THE COMPILER 00000010
%# SUMMARY, I.E. 7700, 7/16/74 DHB 00000020
INTEGER TOKENS; % NUMBER OF SYNTACTIC ITEMS SCANNED, 00504175
  TOKENS:=TOKENS+1; % COUNT OF SYNTACTIC ITEMS PROCESSED 02909500
  "PROGRAM CONTAINS",I7," SYNTACTIC ITEMS,"/
  TOKENS, % SYNTACTIC ITEM COUNT 09274455
%# PATCH NUMBER 501 FOR XALGOL CONTAINS 3 CARDS 09419250
%# THIS IS AN IMPROVED VERSION OF PATCH XIV,105. IT COMPILES A FILE 14155900
%# DECLARATION SUCH AS FILE FILE 3"ABCDE" ,,WITHOUT 14156100
%# MAKING THE USE OF OCTAL AND HEX STRINGS IMPOSSABLE IF THE 14156300
%# LAST THING ENTERED INTO THE SYMBOL TABLE WAS A FILE, 12/20/72 14156500
%# VOID 02761502 % VOID XIV,105 14156700
  IF STOPENTRY THEN DEFINECTR:=99; % STOP OCTAL HEX PROBLEM 02761500
  IF DEFINECTR>95 THEN DEFINECTR:=0; % RESET DEFINCTR 13080750
  13081250

```

NOT IN UCSC

```

$# PATCH NUMBER 800 FOR XALGOL CONTAINS 35 CARDS
$! THIS PATCH ADDS THE CAPABILITY TO SYNTAX
$! THE VARIOUS DCP CONSTRUCTS
$! ERRORBRANCH ATTRIBUTE ADDED 6/5/72
$! THIS IMPLEMENTS THE COMPILER PORTION OF THE XWRITE ATTRIBUTE.11/24/72

```

NOT IN CCSC

```

28 *
,"#LASTS" % LASTSTATION 08493094
,"#+0#LASTS" % LASTSTATUS 08493121
,"#7TIMEO" % TIMEOUT 08493122
,"#8CHART" % CHARTYPE 08493123
,"#8DCACC" % DCACCESS 08493124
,"#5DCUSE" % DCUSE 08493125
,"#H0BUSERC" % USERCODE 08493126
,"#+09STNST" % STNSTATUS 08493127
,"#805XREAD" % EXTENDED READ 08493128
,"#80#ERROR" % ERRORBRANCH 08493129
,"#806XWRIT" % EXTENDED WRITE FOR SYN LINES 08493130
) % END OF FILL STATEMENT (WITH DCP STUFF SAFELY HIDDEN) 08493131
DEFINE NBATTRIBUTES = % THE NUMBER OF STANDARD BIG B ATTRIBUTES 08493135
17# % MARK XV,2,0 2/13/74 08493501
DCPOFFSET = % THE NUMBER OF OVER-LAPPED ATTRIBUTES 08493502
5# % MARK XV,2,0 2/13/74 08493503
DCPATRIBUTENDX= (ATTRIBUTEINDX - DCPOFFSET)#; 08493504
EMITNUM(IF ATTRIBUTEINDX= 1 THEN "6ACCESS" ELSE 08493790
IF ATTRIBUTEINDX= 4 THEN "60THRUS" ELSE 08493791
IF ATTRIBUTEINDX=12 THEN "6ARASIZ" ELSE 08493792
IF DCPATRIBUTENDX=13 THEN "6LSTSTN" ELSE 08493793
IF DCPATRIBUTENDX=14 THEN "6LSTSTS" ELSE 08493794
IF DCPATRIBUTENDX=15 THEN "6TIMOUT" ELSE 08493795
IF DCPATRIBUTENDX=16 THEN "6CHRTP" ELSE 08493796
IF DCPATRIBUTENDX=17 THEN "6DCACES" ELSE 08493797
IF DCPATRIBUTENDX=20 THEN "6STNSTS" ELSE 08493798
IF DCPATRIBUTENDX=22 THEN "6ERRBRN" ELSE 08493799
IF DCPATRIBUTENDX=23 THEN "6XWRITE" ELSE 08493800
0 & FILEATTRIBUTES[ATTRIBUTEINDX][6:12:36] 08493820
& FILEATTRIBUTES[ATTRIBUTEINDX][1:3:11] ) 08493830
EMITL((ATTRIBUTEINDX = % 08493840
(IF ATTRIBUTEINDX > NBATTRIBUTES THEN DCPOFFSET+1 08493842
ELSE 1) 08493844
) & REAL(N#FP OR N#FA)[39:47:11] 08493846

```

LABEL 00000000LINE 00178104?EXECUTE 0/R

0 /R

?EXECUTE 0/R

PACKET 31
INPUT 6 CARDS FROM CRA
TIME 1134
DATE 78104 FRIDAY, 04/14/78

*** BURROUGHS B5700 DCMCP MARK XVI,0,178 AND INTRINSICS MARK XVI,8,132 ***

#NO MESSAGES TODAY

11134157 ?EXECUTE 0/R
11134158 ?COMMON= 1
11134158 ?FILE LINE= LINE PRINT
11134158 ?FILE TTY= TTY DUMMY
11134158 ?FILE S= XALGOL/CUSTOM
11134158 ?END
11134159 410/R= 1 BOJ 1134 11/22/76
11135100 PBD0032 OUT 011 LINE10/R= 1
11135100 DKA IN SER XALGOL CUSTOM10/R= 1
11135132 DKA REL XALGOL CUSTOM10/R= 1
11135132 PBD0032 REL 011 LINE 158410/R= 1
11135132 0/R= 1 EOJ 1135
11135133 FOR 0/R= 1: PROCESS= 19 SECS, IO= 32 SECS, OLAY= 0
11135133 PKT#0031 REMOVED

```

* PATCH 400 FOR XALGOL, XIII CONTAINS 65 CARDS 09417120
COMMENT ***** 00490010
***** 00490020
DESCRIPTION OF NCSL PATCHES 00490030
5 / 5 / 72 00490040
***** 00490050
***** 00490060
THESE PATCHES ARE INCLUDED TO PROVIDE IMPROVEMENTS 00490070
IN BOTH EFFICIENCY AND USER CONVENIENCE, THEY HAVE BEEN 00490080
COLLECTED FROM A VARIETY OF SOURCES, PRINCIPALLY FROM THE 00490090
WESTINGHOUSE RESEARCH LABORATORY, 00490100
----- 00490110
401 NSRDL*PC 00490120
THIS PATCH REDUCES THE SIZE OF THE DISK AREAS CREATED BY THE 00490130
COMPILER AND SETS THE SAVE TIME OF THE NEWTAPE FILE TO 00490140
15 DAYS AND THE LINE FILE TO 1 DAY, 00490150
----- 00490160
402 NSRDL*PC 1/4/71 DHB 00490170
IN ORDER THAT THE LINE PRINTER FILE WILL NOT BE LOST IF IT 00490180
IS LABELQUATED TO DISK AND THE COMPILER IS DS'ED THIS 00490190
PATCH CAUSES SUCH A LINE FILE TO BE LOCKED AND ENTERED IN 00490200
THE DISK DIRECTORY WHEN THE COMPILER HEADING IS WRITTEN 00490210
BY THE DATIME PROCEDURE, 00490220
----- 00490230
403 NSRDL*PC 1971 00490240
THIS PATCH MODIFIES THE COMPILER LISTING AS FOLLOWS: 00490250
A. A SINGLE SPACED LISTING IS ASSUMED BY DEFAULT AND 00490260
THE S OPTION "SINGLE" MUST BE RESET TO GIVE 00490270
DOUBLE SPACING, 00490280
B. THIS ADDS THE ACTUAL PROCESSING TIME USED BY THE COMPILER 00490290
TO THE SUMMARY AND REMOVES THE ESTIMATED AUXILIARY 00490300
MEMORY REQUIRED, 00490310
C. THIS SETS THE COMPILERS ERROR LIMIT VALUE TO A DEFAULT 00490320
VALUE OF 25. THIS VALUE MAY BE CHANGED AT COMPILE TIME 00490330
BY THE "LIMIT" S OPTION, 00490340
----- 00490350
404 WR 00490360
THIS PATCH MODIFIES AND CORRECTS THE COMPILERS METHOD OF 00490370
CALCULATING A PROGRAMS CORE ESTIMATE. 00490380
----- 00490390
405 WR, NSRDL*PC DAVE BROWN 00490400
THIS PATCH IMPROVES THE COMPILERS EFFICIENCY BY 00490410
PROVIDING ADDITIONAL SEGMENTATION AND BY REWRITING 00490420
SOME POOR CODING, 00490430
----- 00490440
406 DAVE BROWN 5/5/72 00490450
THIS ALLOWS 8 CHARACTER BCL, 16 OCTADE OCTAL AND 12 HEXADE HEX 00490460
STRINGS SO LONG AS THE FLAG BIT IS NOT SET, 00490470
----- 00490480
407 WR 00490490
THIS IMPLEMENTS THE STORE*STAR OPERATOR OF THE FORM, 00490500
A[I;J] := * + AEXP OR 00490510
BEJ := (*) & AEXP [CONCAT], ETC. 00490520

```

19121
MOORE BUSINESS FORMS, INC. NY

THE STAR (*) AS THE FIRST OPERAND AFTER THE REPLACEMENT OPERATOR CAUSES A POLISH(DUP,LOD) TO BE EMITTED INSTEAD OF RE-EVALUATING THE ENTIRE SUBSCRIPT EXPRESSION,

00490530
00490540
00490550
00490560

408 NSRDL=PC 7/25/71 DHB
TWO NEW FOR=LIST TYPES ARE IMPLEMENTED BY THIS PATCH,
THE FOLLOWING TABLE GIVES EXAMPLES OF THE NEW FOR=LIST
TYPES AND THEIR EQUIVALENT ALGOL-60 CONSTRUCTS,
NEW FOR=LIST TYPE ALGOL-60 EQUIVALENT

00490570
00490580
00490590
00490600
00490610
00490620
00490630
00490640
00490650

A(B)C A STEP B UNTIL C
A/C A STEP 1 UNTIL C

***** END OF NCSL PATCH DESCRIPTIONS *****

\$* PATCH 401 FOR XALGOL,XIII CONTAINS 3 CARDS
DEFINE CHUNK = 90;
FILE OUT LINE DISK SERIAL [201300] (RR3,15,RR4,SAVE 1);
SAVE FILE OUT NEWTAPE DISK SERIAL [201600] (RR5,RR6,RR7,SAVE 15);
\$* PATCH 402 FOR XALGOL,XIII CONTAINS 6 CARDS
IF Q1=LINE,TYPE=12 OR Q=10 OR Q=13 THEN % GOING TO DISK

01556100
01559000
01560000

BEGIN
N1:=1; % DONT NAME FILES FOR REMOTES
LINE,TYPE:=12; WRITE(LINE); LOCK(LINE); % SO ITS NOT LOST ON DS
LINE,AREAS:=0; LINE,AREASIZE:=0;

01828800
01828810
01828820
01828830
01828840
01828850

END
\$* PATCH NUMBER 403 FOR XALGOL CONTAINS 91 CARDS
\$* MAXSEG AND IMPROVED SUMMARY ADDED 2/13/74, DHB,
INTEGER MAXSEG; % FOR LARGEST SEGMENT FOUND,
ARRAY LIN[0:9];

00000000
00 00010
01693100

STREAM PROCEDURE CODENAME(N1,N2,LIN); VALUE N1,N2;
BEGIN % C O D E / N A M E

01822500
01827010
01827020

LOCAL N;
SI:=LOC N; SI:=SI+1;
7(IF SC="" THEN JUMP OUT; TALLY:=TALLY+1; SI:=SI+1);
N:=TALLY; % NUMBER OF CHARS IN FIRST NAME
DI:=LIN; DS:=LIT""; SI:=LIN; DS:=B WDS;
DI:=LIN; DI:=DI+52; % TO THE "/"
DI:=DI-N; DI:=DI-N;
SI:=LOC N; SI:=SI+1;
N(DS:=CHR; DI:=DI+1);
DS:=LIT"/"; DI:=DI+1;
SI:=LOC N; SI:=SI+1;
7(DS:=CHR; DI:=DI+1);

01827030
01827040
01827050
01827060
01827070
01827080
01827090
01827100
01827110
01827120
01827130
01827140
01827150

END;
STREAM PROCEDURE UNDERLINE(LIN);

01827160

BEGIN
SI:=LIN; SI:=SI+38;
DI:=LIN; DI:=DI+38;
15(IF SC="" THEN DI:=DI+1 ELSE DS:=LIT""; SI:=SI+2; DI:=DI+1);

01827170
01827180
01827190
01827200
01827210

END;
STREAM PROCEDURE NAMEIT(A,B,W,LIN); VALUE A,B,W;

01827220
01827230

BEGIN
LABEL ZER,ONE,NAM,FILE; LOCAL N;
DI:=LIN; DI:=DI+37; SI:=LOC A; SI:=SI+1;
CI:=CI+W; GO TO ZER; GO TO ONE; % GO TO TWO;
DS:=12LIT"NEW SYMBOLIC"; GO TO NAM;
ONE: DS:= 6LIT"PATCH "; GO TO FILE;

01827240
01827250
01827260
01827270
01827280
01827290

Moore Business Forms, Inc. 5/71


```

ZER: DSI= 6LIT"SOURCE"; 01827300
FIL: DSI= 6LIT" FILE "; 01827310
NAM: DSI= 3LIT" JS"; DSI=8LIT" "; 01827320
7(IF SE=" " THEN JUMP OUT; TALLY:=TALLY+1; SI:=SI+1); 01827330
NI=TALLY; SI=SI-NI; DI=DI-NI; DSI=N CHR; 01827340
DSI=LIT"/"; SI:=LOC B; SI:=SI+1; DSI:=7 CHR; 01827350
END; 01827360
ELSE 01828860
BEGIN % GIVE LINE FILE NAME OF CODE FILE 01828870
IF STATUS(LIN[*]) > 0 THEN LINE,TYPE:=15; % REMOTE INITIATION 01828880
LINE,MFID:=NI; LINE,FID:=N2; 01828890
END; 01828900
WRITE(LINE[DBL], % 01829000
" " ,A6,"DAY , ",0," , ",12," ",A2,X1,A3>, 01832000
$ VOID 01832500
IF H ≥ 12 THEN "PM," ELSE "AM,"); 01835000
$ VOID 01835001
IF N1 ≥ 0 THEN % NOT FOR REMOTE SO NAME HIS FILES 01835025
BEGIN 01835050
WRITE(LINE[DBL]); 01835075
CODENAME(N1,N2,LIN); WRITE(LINE,9,LIN[*]); 01835100
UNDERLINE(LIN); WRITE(LINE[DBL],9,LIN[*]); 01835125
NI:=CARD,MFID; N2:=CARD,FID; 01835150
IF MERGETOG THEN % $ TAPE 01835175
BEGIN 01835200
NAMEIT(TAPE,MFID,TAPE,FID,0,LIN); 01835225
WRITE(LINE[DBL],9,LIN[*]); 01835250
END; 01835275
IF CARD,TYPE=12 THEN 01835300
NAMEIT(N1,N2,REAL(MERGETOG),LIN) 01835325
ELSE 01835350
NAMEIT("CARD " , "DECK " ,REAL(MERGETOG),LIN); 01835375
WRITE(LINE[DBL],9,LIN[*]); 01835400
IF NEWTOG THEN % $ NEWTAPE 01835425
BEGIN 01835450
NAMEIT(NEWTAPE,MFID,NEWTAPE,FID,2,LIN); 01835475
WRITE(LINE[DBL],9,LIN[*]); 01835500
END; 01835525
WRITE(LINE); 01835550
END; 01835575
OPTIONWORD:=FALSE&TRUE(SINGLBIT); % SET SINGLE RESET REST 02130000
OPTIONS[43]:=1; % FOR SINGLE SPACE TO BE DEFAULT 02131100
"6SINGL",1, % 42, 43 09027044
SINGLTOG:=TRUE; % DEFAULT SINGLE SPACE LISTING 09028020
ERRMAX:=25; % DEFAULT VALUE, USER CAN CHANGE BY $ CARD 09828910
FORMAT PAN ( % COMPILER SUMMARY 09274400
"NUMBER OF ERRORS DETECTED ="14", COMPILE TIME ="15" IN "16" SECONDS," 09774410
X16,2A4,/, 09774420
"PROGRAM SIZE! CARDS PRT DISK SEGS PGM SEGS MAX SEG " 09274430
"WDS OF CODE"/ 09774440
X14,15,17,110,111,110,112// 09774450
"ESTIMATED CORE STORAGE REQUIRED ="16," WORDS,"); 09774460
WRITE(LINE,PAN, % COMPILER SUMMARY 09417000
ERRORCOUNT,TIME(2)/60,(TIME(1)*TIME1)/60,GT1,GT2, 09417500
CARDCOUNT,PRTIMAX, 09418000
IF DASCHUNK THEN DA ELSE *CHUNK*ENTIER(*DA/CHUNK), 09418500
SGAVL=1,MAXSEG,AKKUM, 09419000

```

MOORE BUSINESS FORMS, INC. NY

```

GT11))
END)
IF SIZE>MAXSEG THEN MAXSEGI=SIZE; % LARGEST SO FAR
$# PATCH 404 FOR XALGOL.XIII CONTAINS 7 CARDS
$ VOIDT
GT11:=IF NOOFARRAYS > 10 THEN 5000 ELSE NOOFARRAYS*500;
$ VOID 09361061
GT11:=GT11+CIF AKKUM > 10000 THEN 5000 ELSE AKKUM/2); %
IF (IOTEMP:=GET(L=2)),[46;2] = 0 THEN CURRENT:=IOTEMP
IF (IOTEMP:=GET(L=1)),[46;2] = 0 THEN IOTEMP:=IOTEMP DIV 4
$# PATCH 405 FOR XALGOL.XIII CONTAINS 269 CARDS
DEFINE EFFSEGMENT =DEFINE ZOTSEGHHERE=##; % TO FORCE A SEGMENT
$ VOID 01339051
DEFINE FSAVE=RR10#; COMMENT SAVES FRACTIONAL PART EXPONENT ;
INTEGER IDLOC; % POINTER INTO IDARRAY (FILE PARAMETER BLOCK)
$ VOID 01419001
$ VOID 01525001
INTEGER KOUNT;
$ VOID 01601001
$ VOID 01687001
$ VOID 0173725#
$ VOID 01786001
DEFINE LOOKDEC=% USED ONLY IN SEARCHLIB, SEE 02013184
%; % END OF LOOKDEC DEFINE
LOOKDEC % DECLARS STREAM PROCEDURE LOOK SEE 2001010
SVOID 02023000
$ VOID 02263001
; LOCK(POP[*]); LOCK(WOP[*]);
COUNT:=0; T:=63; %
$ VOIDT
$ VOIDT
$ VOID 03018001
$ VOID 03033001
DEFINE CONSTANTCLEAN=BEGIN IF MRCLEAN THEN CONSTANTCLEANP END#;
PROCEDURE CONSTANTCLEANP; FORWARD;
DEFINE PASSTYPE(E)=((E,CLASS=BOOPROCID),[46;2]);#;
DEFINE SORTSTMT=SORTMERGESTMT(FALSE)#;
MERGESTMT=SORTMERGESTMT(TRUE)#;
PROCEDURE SORTMERGESTMT(MERGE); VALUE MERGE; BOOLEAN MERGE; FORWARD;
BEGIN EFFSEGMENT;
PROCEDURE CONSTANTCLEANP;
$
BEGIN EFFSEGMENT;
END B2D;
SVOID 04277000
BEGIN EFFSEGMENT;
BUGGER(S,[36;10]) ELSE WOPIT1;IF T1=1 THEN RR10
BUGGER(S,[36;10]) ELSE WOPIT1;IF T1=1 THEN RR10
END EFFSEGMENT DEBUG;
COMMENT PACK IS A STREAM PROCEDURE WHICH INSERTS THE SYLLABLE
INTO THE EDOC ARRAY, THE SPECIFIC ELEMENT OF EDOC
IS PRECISLY = EDOC((L DIV 4) DIV 128+(L DIV 4)MOD 128)
SYLLABLE POSITION=(L MOD 4), WHERE L IS THE SYLLABLE
NUMBER RELATIVE TO THE BEGINNING OF THE SEGMENT;
STREAM PROCEDURE PACK(WORD,POSITION,SYLLABLE);
VALUE POSITION,SYLLABLE;

```

```

09419500
09419750
13681100
09361005
09161040
09361050
09161080
09361100
13113000
413148200
01800060
01339050
01340500
01341000
01416500
01525000
01583000
01598000
01684000
01537000
01786000
02801010
02801610
02813184
02020000
02262000
02562000
02491000
02716000
02904000
03018000
03031000
03034000
03834100
03874000
03081000
03082000
03082100
04143000
04163000
04164000
04247090
04248010
04265000
04278100
04281000
04286000
04287100
04292100
04292110
04292120
04292130
04292140
04292150
04292160

```

```

BEGIN
DI*WORD ; DI ← DI+POSITION ; DI ← DI+POSITION;
ST*LOC SYLLABLE ; SI*SI+6;
DS*2 CHR ;
END PACK ;
RR101=" " ;
RR101:=(IF Q=10 THEN OP,[3913]&OP[41138,1] ELSE
DEFINE PUNCH(FL,ST)=IF WRITNEW(FL,ST) THEN#;
$ VOID 05056001
$VOID 05101000
BEGIN EFFSEGMENT; FLAG(ERRNUM);
EFFSEGMENT;
BOOLEAN RRB2;
DEFINE CHECKPRESENCE = % FOR EFF
END CHECKPRESENCE;#;
BEGIN EFFSEGMENT;IF SPAC=0 THEN SPAC←GETSPACE(TRUE,-2);
EFFSEGMENT;
BEGIN EFFSEGMENT; RANGE←ELCLASS≥LOWER AND ELCLASS≤UPPER END;
DEFINE CALLSWITCH(H)= %
#;
$ VOID 05325001
$ VOID 05387001
$VOID 05407001
DEFINE IFCLAUSE = %
BEGIN STEPIT; STACKCT ← 0; BEXP;
IF ELCLASS ≠ THENV THEN ERR(116)ELSE STEPIT END IFCLAUSE;#;
$ VOID 06411001
EFFSEGMENT;
EFFSEGMENT;
BEGIN
HOLE ← 2 = REAL(FROM);
VARIABLE(HOLE);
GO TO EXIT
END;
EFFSEGMENT;
DEFINE C=RR1#; % TO HOLD L FOR A WHILE
BEGIN
DEFINE C=N#;
BEGIN
$ VOIDT
I←I+1; STEPIT; %
MOVECODE(TEDOC,EDOC); %
$ VOID 08276001
LOCK(FILEATTRIBUTES[*]); I←FILEATTRIBUTES[0];
%
PROCEDURE SORTMERGESTMT(MERGE); VALUE MERGE; BOOLEAN MERGE;
BEGIN INTEGER J,K,FILEL,FILEND; LABEL QUIT; BOOLEAN OPTOG,INPRO;
DEFINE A=K#, RDS=1,280#, OUTPRO=OPTOG#;
COMMENT ARRAYCHECK CHECKS A PARAMETER=INFO WORD FOR SORT/MERGE;
BOOLEAN PROCEDURE ARRAYCHECK(AAW); VALUE AAW; REAL AAW;
ARRAYCHECK←AAW.CLASS<BOOARRAYID OR AAW.CLASS>INTARRAYID
OR AAW,INCR #1;
COMMENT COMMACHECK LOOKS FOR COMMAS AND STEPS AROUND THEM;
BOOLEAN PROCEDURE COMMACHECK;
BEGIN IF NOT(COMMACHECK←(STEPI=COMMA)) THEN ERR(350);
STEPIT
END COMMACHECK;

```

```

04292170
04292180
04292190
04292200
04292210
04602000
04406000
05831100
05050000
05061000
05106000
05152500
05166100
05184000
05187000
05760000
05293100
05104000
05315000
05316100
05317000
05379000
05388000
06291100
%A 06291200
06291300
06291400
06409000
06414100
06420100
07402000
07402100
07402200
07402300
07402400
07538100
07596100
07609200
07646397
07646760
07689000
07691000
07698500
08246000
08493140
08901000
08901200
08902000
08902010
08902020
08902030
08902040
08902050
08902060
08902070
08902080
08902090
08902100

```

```

COMMENT HVCHECK CHECKS VALIDITY OF HIVALU PROCEDURE FOR SORT;
BOOLEAN PROCEDURE HVCHECK(ELBW); VALUE ELBW; REAL ELBW;
  IF ELBW,CLASS#PROCID THEN ERR(356) ELSE
  IF BOOLEAN(ELBW,FORMAL) THEN HVCHECK*TRUE ELSE
  IF TAKE(GTI*GIT(ELBW))*1 THEN ERR(357) ELSE
  IF ARRAYCHECK(TAKE(GT1+1)) THEN ERR(358) ELSE
  HVCHECK*TRUE;
COMMENT OUTPROCHECK CHECKS SORT/MERGE OUTPUT PROCEDURE;
BOOLEAN PROCEDURE OUTPROCHECK(ELBW); VALUE ELBW; REAL ELBW;
  IF ELBW,CLASS#PROCID THEN ERR(351) ELSE
  IF BOOLEAN(ELBW,FORMAL) THEN OUTPROCHECK*TRUE ELSE
  IF TAKE(GT1*GIT(ELBW))*2 THEN ERR(352) ELSE
  IF TAKE(GT1+1),CLASS#BOOID THEN ERR(353) ELSE
  IF ARRAYCHECK(TAKE(GT1+2)) THEN ERR(354) ELSE
  OUTPROCHECK*TRUE;
COMMENT EQLESCHECK CHECKS THE COMPARE ROUTINE FOR SORT/MERGE;
BOOLEAN PROCEDURE EQLESCHECK(ELBW); VALUE ELBW; REAL ELBW;
  IF ELBW,CLASS#BOOPROCID THEN ERR(359) ELSE
  IF BOOLEAN(ELBW,FORMAL) THEN EQLESCHECK*TRUE ELSE
  IF TAKE(GTI*GIT(ELBW))*2 THEN ERR(360) ELSE
  IF ARRAYCHECK(TAKE(GT1+1)) THEN ERR(361) ELSE
  IF ARRAYCHECK(TAKE(GT1+2)) THEN ERR(362) ELSE
  EQLESCHECK*TRUE;
STREAM PROCEDURE STUFFILE(IDLOC*FN; SFN);
  VALUE FN; SFN;
BEGIN DI*IDLOC; DI*DI+5; DI*DC;
  SI*LOC FN; SI*SI+5; DS*3 CHR; SI*SI+7;
  DS*11 LIT"000000DSRT"; DS*CHR; SI*SI-1;
  DS*7 LIT"SDSRT"; DS*CHR; SFN*DI; SI*LOC SFN;
  DI*IDLOC; DI*DI+5; SI*SI+5; DS*3 CHR;
END STUFFILE;
BOOLEAN PROCEDURE INPROCHECK(ELBW); VALUE ELBW; REAL ELBW;
  IF ELBW,CLASS#BOOPROCID THEN ERR(363) ELSE
  IF BOOLEAN(ELBW,FORMAL) THEN INPROCHECK*TRUE ELSE
  IF TAKE(GTI*GIT(ELBW))*1 THEN ERR(364) ELSE
  IF ARRAYCHECK(TAKE(GT1+1)) THEN ERR(365) ELSE
  INPROCHECK*TRUE;
IF MERGE THEN
  BEGIN % MERGE
  END MERGE
ELSE
  BEGIN % SORT
  $VOID 08945000
  END SORT;
END SORT AND MERGE STATEMENTS;
$ VOID 09024000
$ VOIDT
PROCEDURE GETREADY;
  BEGIN EFFSEGMENT;
  $ VOIDT
  $ VOIDT 09028073
  $ VOID 09056001
  LOCK(TENT*)); % READ ONLY
  LOCK(SPECIAL*)); % READ ONLY
  LOCK(SUPERSTACK*)); % READ ONLY
  GTI*GETSPACE(TRUE,*5); ERRORTOGI*TRUE;
END GETTING READY;

```

```

08902200
08902210
08902220
08902230
08902240
08902250
08902260
08902270
08902280
08902290
08902300
08902310
08902320
08902330
08902340
08902350
08902360
08902370
08902380
08902390
08902400
08902410
08902420
08902430
08902440
08902450
08902460
08902470
08902480
08902490
08902500
08902510
08902520
08902530
08902540
08902550
08902560
08903000
08904000
08926000
08927000
08928000
08929000
08996100
08998000
09003000
09024000
09025000
09025100
09027000
09828068
09042000
09251310
09251320
09251350
09774000
09774010

```

```

PROCEDURE CLEANUP;
BEGIN LABEL L1;
STREAM PROCEDURE MDESC(WD, TOLOC); VALUE WD;
BEGIN DI←LOC WD; DS← SET; SI← LOC WD; DI←TOLOC; DS←WDS END;
COMMENT THE FOLLOWING PROCEDURE PRINTS OUT THE PRT, NAME, AND
SEGMENT NUMBER OF THE INTRINSIC PROCEDURES USED IN THE
OBJECT PROGRAM;
STREAM PROCEDURE WRTINTRSC(SGNO, ALFA, PRT, FIL);
VALUE SGNO, PRT;

BEGIN LOCAL COUNT, DEST;
DI←FIL; DS←4 LIT"PRT("); SI←LOC PRT; SI←SI+4; TALLY←4;
3(CIF SC="0" THEN % DONT PRINT LEADING ZEROES,
BEGIN SI←SI+1; TALLY←TALLY+63 END ELSE JUMP OUT);
COUNT←TALLY; DS←COUNT CHR; DS←4 LIT") = ";
SI←ALFA; SI←SI+2; DEST←DI; % SAVE DI,
DI←LOC COUNT; DS←7 LIT"0"; DS←CHR; % NO OF CHARS IN NAME,
DI←DEST; DS←COUNT CHR; % INT, NAME,
DS←29 LIT" INTRINSIC, SEGMENT NUMBER = ";
SI←LOC SGNO; DS←4 DEC; DS←LIT", ";
DI←DI+5; DS←4 FILL; % JUNK LEADING BLANKS,
END WRTINTRSC;

DEFINE STARTINTRSC=426#;
ARRAY PRT, SEGDICT[0:7, 0:127];
INTEGER PRTADR, SEGMENT, LINK;
INTEGER PROCEDURE MOVEANDBLOCK(FROM, SIZE); VALUE SIZE;
ARRAY FROM[0, 0]; INTEGER SIZE;
BEGIN INTEGER NSEGS←I÷J;
REAL T;
NSEGS←(SIZE+29) DIV 30;
IF DA DIV CHUNK<T*(DA+NSEGS) DIV CHUNK THEN
DA←CHUNK×T;
MOVEANDBLOCK←DA;
DO BEGIN FOR J←0 STEP 2 WHILE J<30 AND I<SIZE DO
BEGIN MOVE(2, FROM[I DIV 128, I MOD 128], CODE(J));
I←I+2; END;
WRITE(CODE[DA]); DA←DA+1;
END UNTIL I≥SIZE;
END MOVEAND BLOCK;
STREAM PROCEDURE ZEROUT(DEST, NDIV32, NMOD32);
VALUE NDIV32, NMOD32 ;
BEGIN DI I= DEST;
NDIV32(32(CDS I=8 LIT"0"));
NMOD32(DS I= 8 LIT"0");
END;
$ VOID 09333000
$ VOID 09348000
GT1←MIN((IDLOC-MKABS(IDARRAY[0])), [33!15]+1, 128); %
BEGIN
$ VOID 09415000
END OF CLEAN UP;
***** ACTUAL COMPILATION DONE HERE *****
DEFINE PROGRAM =
GETREADY;
BLOCK(FALSE);
CLEANUP;
#;
*****
$

```

```

09774020
09774030
09274040
09774050
09774060
09774070
09274080
09774090
09274100
09774110
09774120
09274130
09774140
09774150
09274160
09774170
09274180
09774190
09274200
09274210
09774220
09774230
09774240
09774250
09774260
09274270
09774280
09274290
09274300
09274310
09274320
09274330
09274340
09274350
09274360
09274370
09274380
09774390
09275000
09275010
09275020
09275030
09275040
09275050
09317000
09334000
09394000
09409000
09410000
09420000
09421000
09421100
09422000
09423000
09424000
09424100
09425000
09426000

```

```

EFFSEGMENT; 10070000
COMMENT GETINT DOES A CALL ON NEXTENT AND CHECKS TO SEE IF AN INTEGER 10105005
WAS THE RESULT; IF NOT ERROR = OTHERWISE MAKE SIGN PLUS; 10105010
BOOLEAN PROCEDURE GETINT; 10105020
BEGIN NEXTENT; IF ELCLASS * * ELCLASS < 0 THEN 10105030
IF ELCLASS=("(**") THEN GETINT+TRUE ELSE 10105040
BEGIN FLAG(137); ELCLASS * 0 END 10105050
END GETINT; 10105060
COMMENT DIVIDE PARTIONS THE PARAMETER NUMBER INTO THREE PARTS, THE 10105070
RESULT IS PASSED BACK THROUGH P1,P2, AND THE FUNCTION 10105080
IDENTIFIER, SEE CODE FOR DETAILS; 10105090
INTEGER PROCEDURE DIVIDE(NUMBER,P1,P2); 10105100
VALUE NUMBER; INTEGER P1,P2,NUMBER; 10105110
BEGIN 10105120
IF NUMBER < 0 THEN BEGIN FLAG(138); ERRORTOG+TRUE; 10105130
NUMBER * 0 END; 10105140
P1 * IF NUMBER < 8 THEN NUMBER ELSE 8; 10105150
NUMBER * NUMBER=P1; 10105160
P2 * IF NUMBER < 8 THEN NUMBER ELSE 8; 10105170
DIVIDE * NUMBER=P2 END DIVIDE; 10105180
$ VOID 10193000 10175000
DEFINE PACKINFO(INFO,ISKIP,COUNT,ASKIP,ACCUM)= 10233100
MOVECHARACTERS(COUNT,ACCUM,ASKIP+3,INFO,ISKIP)#; 10233200
$ VOID 10240001 10236000
$ VOID 13144001 13115000
$ VOID 13210001 13209000
BEGIN EFFSEGMENT; 13369000
$ VOID 13537001 13532000
PUT(X,NEXTINFO); * FOR EFF,.. 13593000
BEGIN EFFSEGMENT; 13632100
DEFINE MOVE2(S,D)=MOVE(2,S,D)#; 13474000
$ VOID 13676001 13675000
$ VOID 13682001 13682000
EFFSEGMENT; 13715100
BEGIN HANDLESWLST; *SAVE 14168000
GOTSTORAGE * NOT SPECTOG OR GOTSTORAGE; *PRT 14168100
GO TO START END; *CELL 14168200
DEFINE CHKSQB=IF GTAIL(J)=J*11#0 THEN FLAG(23)#; * 14273100
$ VOID 14273201 14273200
EFFSEGMENT; 14527000
EFFSEGMENT; 15006100
$ VOID 15019001 15014000
REAL STREAM PROCEDURE GETALPHA(INFOINDEX,SIZE); 15830100
VALUE SIZE; 15030200
BEGIN COMMENT GETALPHA PICKS ALPHA CHARACTERS OUT OF INFO AND 15830300
FORMATS THE ID WORD THAT IS PASSED TO PRINTI, THE FIRST 15030400
CHARACTER CONTAINS THE SIZE, THE NEXT CHARACTER CONTAINS THE 15030500
ALPHA BEFT JUSTIFIED WITH TRAILING ZEROS; 15030600
DI+LOC GETALPHA; DS+8 LIT"0 " DI+DI-7; 15030700
SI+INFOINDEX; SI+SI+3; DS+SIZE CHR; 15030800
END GETALPHA; 15030900
$# PATCH NUMBER 406 FOR XALGOL CONTAINS 31 CARDS
$! THIS MAKES 8 CHARACTER BCL, 16 OCTADE OCTAL AND 12 HEXADE HEX STRINGS 09417110
$! ACCEPTABLE SO LONG AS THE FLAG BIT IS NOT SET, 5/15/72 DHB 02576030
$! THIS CHANGES THE WAY THE COMPILER TREATES OCTAL AND HEX STRINGS, 02847005
$! THEY ARE NOW CONSIDERED AS A STRING OR STRINGCON AS OTHER STRINGS 02847010
$! THIS DOES NOT CHANGE THE WAY ALGOL USES THESE CONSTANTS BUT 02847015

```

Moore Business Forms, Inc. NY

```

$! IN XALGOL THE STATEMENTS;                                02847020
$!   REPLAC . . . BY "A";   AND                               02847025
$!   REPLAC . . . BY 3"21";                                   02847030
$! WILL GENERATE THE SAME RESULT,   1/16/74,
PROCEDURE DOOCTALORHEXSTRING;                                02001855
BEGIN LABEL AWAY; %                                          02001856
    RESULT:=5; SCANNER; % SKIP QUOTE,                        02001900
    COUNT:=0; GT1:= 48 DIV C; % 16 OR 12                     02001905
    DO BEGIN                                                  02001910
        RESULT:=5; SCANNER;                                   02001915
        IF COUNT > GT1 THEN % > 1 WORD LONG,                02001920
            BEGIN ERR(520); GT1:=0; GO AWAY END;             02001925
        END UNTIL EXAMIN(NCR)="";                             02001930
        Q:=ACCUM[1]; RESULT:=5; SCANNER; COUNT:=COUNT-1;    02001935
        IF C=3 THEN % OCTAL STRING,                            02001940
            IF OCTIZE(ACCUM[1],ACCUM[4],16=COUNT,COUNT) THEN 02001945
                FLAG(521) % NON-OCTAL CHARACTER IN STRING,  02001950
            ELSE ELSE IF HEXIZE(ACCUM[1],ACCUM[4],12=COUNT,COUNT) THEN 02001955
                FLAG(521) % NON-HEX CHARACTER IN HEX STRING, 02001960
                COUNT:=(COUNT*C+C) DIV 6; % NUMBER OF 6 BIT CHAR IN STRING 02001965
                MOVECHARACTERS(COUNT,ACCUM[4],8=COUNT,ACCUM[1],3); 02001980
                GT1:=1; Q:=ACCUM[1];=(*)&COUNT[12;42;16]; 02801985
            AWAY; %                                            02001990
    END OCTAL OR HEX STRING;                                   02001995
    STRNGXT,ARGH, %                                           02438000
    IF COUNT < 8 OR                                           02703000
        COUNT=8 AND NOT BOOLEAN(Q,[18;1]) THEN % NO FBIT STILL OK 02703010
    $ VOID 02704001                                           02704000
        BEGIN %                                               02562000
            DOOCTALORHEXSTRING; %                               02763000
            IF BOOLEAN(GT1) THEN GO TO STRNGXT; % WAS VALID    02764000
        $ VOID 02765001                                         02765000
            GO TO SCANAGAIN; % WAS AN ERROR                    02766000
        $ VOIDT 02782001                                         02767000
        $ VOID 02847001                                         02847000
    $* PATCH 407 FOR XALGOL,XIII CONTAINS 22 CARDS
    IF ELCLASS=CROSSHATCH THEN BEGIN                           06003200
        STEPIT; SIMPARITH END ELSE                             06003400
        INTEGER SL;                                           06089500
        IF ELCLASS=FACTOR THEN                                  06091200
            IF LINKTOG THEN % LAST SYLLABLE IS NOT A LINK    06091300
                IF (SL*GET(L=1)),(46;2)=3 OR SL=673 THEN BEGIN 06091400
                    ELBAT[1],CLASS+ELCLASS+CROSSHATCH; STEPIT; 06091500
                    EMIT(DUP); EMIT(LOD); GO EXIT END;         06091600
                IF ELCLASS=CROSSHATCH THEN GT4=BOOSEC+BTYPE ELSE 06152500
                    IF P1 # FL THEN STEPIT;                    15092015
                    IF T1#0 OR ELCLASS=FACTOR THEN BEGIN EMIT(DUP); 15095000
                        EMIT(COC); IF T1#0 AND ELCLASS=FACTOR THEN 15095500
                            EMIT(DUP) END;                       15095600
                        ELSE IF T1#0 OR ELCLASS=FACTOR THEN BEGIN 15097000
                            EMITV(TALL ) ; IF T1#0 AND ELCLASS=FACTOR 15097500
                                THEN EMIT(DUP) END;             15097600
                            IF ELCLASS=FACTOR THEN BEGIN          893 15098025
                                ELBAT[1],CLASS + ELCLASS +CROSSHATCH END; 893 15098050
                                BEGIN%                             15098200
                                    IF ELCLASS=FACTOR THEN BEGIN 15306200
                                        ELBAT[1],CLASS+ELCLASS+CROSSHATCH; 15306400
                                        EMIT(DUP); IF T1=0 THEN EMIT(LOD) END; 15306600

```

```

$# PATCH 408 FOR XALGOL,XIII CONTAINS 19 CARDS
  LABEL EXIT,GETC) %
  OR ELCLASS=LEFTPAREN OR ELCLASS=COLON
  IF ELCLASS=COLON THEN %
    BEGIN%
      SIMPLEBI=CONSTANBI=TRUE; SIGNBI=FALSE;%
      BI=1; ELCLASS:=UNTILV;%
      PLUG(TRUE,1);%
    END%
  ELSE%
    (STEP1=UNTILV OR ELCLASS=RTPAREN OR ELCLASS=WHILEV)
    OR ELCLASS=RTPAREN %
  BEGIN %% COLON MAYBE
  IF ELCLASS=COLON THEN%
    BEGIN%
      CONSTANBI=TRUE; SIGNBI=FALSE; BI=1; GO GETC %
    END)%
  IF FORCLASS(UNTILV) OR ELCLASS=RTPAREN THEN%
  GETC; IF SIMPLE(CONSTANC,Q,SIGNC) THEN%
  END; %% MAYBE COLON
$# PATCH NUMBER 410 FOR XALGOL CONTAINS 389 CARDS
$! IMPLIMENTS $SYSTEMID, NABS AND NOBITSON, 10/26/72
$! THIS IS THE IPC PATCH FOR ALGOL AND XALGOL, D. H. BROWN---JAN, 1973
  TASKID =115%; COMMENT 163;
  EVENTID =116%; COMMENT 164;
  IPCPROCID =117%; COMMENT 165;
  SIZEV =21%; COMMENT 25;
  TASKV =22%; COMMENT 26;
  EVENTV =23%; COMMENT 27;
  BOOLEAN IPCJOB; % TRUE IF "PROCEDURE" IS FIRST THING TO BE COMPILED,
  % I.E. IF WE ARE COMPILING AN INVOKABLE IPC PROGRAM,
  % AFTER PROCESSING THE IPC PROCEDURE SPECIFICATION THE
  % CF CONTAINS 2xMARK, WHICH POINTS TO THE PARAMETER
  % SPECIFICATIONS IN INFO, THE FF IS USED TO COUNT THE
  % NUMBER OF "PROCESS" STATEMENTS; IF IPCJOB IS NON ZERO
  % AT WRAP UP THEN THE CODE FILE IS MARKED AS INVOKING
  % AND/OR INVOKED AND, IF NEEDED, A PARAMETER DESCRIPTION
  % SEGMENT IS WRITTEN IN THE CODE FILE, 12/13/72 DHBROWN
  POWERSOFTEN =758%;
  LASTSEQUENCE =249%; % 760 - 511
  SORTA =759%;
  <X2;,"CUSTOMS B=5700 IPC XALGOL COMPILER MARK ",
  PROCEDURE TASKATTRIBUTEHANDLER(F); VALUE F; INTEGER F; FORWARD; %0776600003061250
  MAXANDMIN, DELAY, DELTA, OTHERS, SYSTEMID, EXIT;
  MAXANDMIN, MAXANDMIN, DELAY, ABS, % NABS
  SYSTEMID;
  IF BOOLEAN((T1=ELBAT[1]),V0) THEN % WRONG TYPE BOOLEAN
  BEGIN ERR(103); GO TO EXIT END;
  IF T2=T1,[27;6] <9 OR T2=50 THEN
  GO TO STIF T2=50 THEN T2=40 ELSE T2+1;
  ABS; PANAJ EMIT0(IF T2=50 THEN SSN ELSE SSP); GO TO EXIT;
  SYSTEMID; EMITL(7); EMITL(1); EMIT0(COM); STEPIT; GO EXIT; % FPM
  IF ELCLASS=EVENTID AND REALF THEN % REAL(EVENTID) IS OK
  BEGIN EMITV(ELBAT[1]); STEPIT; GO EXIT END; % EVENTS ON STACK
  IF ELCLASS=TASKID THEN %
  BEGIN TASKATTRIBUTEHANDLER(FP); GO TO LAMPER END;
  PROCEDURE BIMPFUN; % HANDLES THE BOOLEAN SPECIAL FUNCTIONS

```

```

08017000
08103100
08111100
08111200
08111300
08111400
08111500
08111600
08111700
08113000
08128100
08182100
08182200
08182300
08182400
08182500
08184000
08185000
08?22100
01778060
01778065
01778070
01298600
01298650
01298700
01401100
01401110
01401120
01401130
01401140
01401150
01401160
01401170
01401180
01574000
01575000
01580000
01830000
00776600003061250
06060100
06061100
06061150
06061500
06061750
06062000
06062050
06067000
06873150
06090178
06090179
06092075
06092077
06210010

```



```

BEGIN % 06210020
LABEL OTHERS, FIX, FREE, AVAILABLE, ACQUIRE, HAPPENED, PEND, NOBITSON, 06210030
EXIT, CKRPN; 06210040
SWITCH % 06210050
FUNCT=OTHERS, FIX, FREE, AVAILABLE, ACQUIRE, HAPPENED, PEND, NOBITSON; 06210060
DEFINE HAPBIT = 8; % BIT [44:1] IS ON IF EVENT HAS HAPPENED, 06210070
AVLBIT = 1; % BIT [47:1] IS ON IF EVENT IS NOT AVAILABLE, 06210080
% IF THE AVLBIT IS ON THEN EVENT, FF CONTAINS 06210090
% THE MIX INDEX OF THE PROCESS IN CONTROL, 06210100
REAL T1, T2; 06210110
06210120
IF 1 ≤ T1=ELBAT[I], [27:6] ≤ 6 THEN % FIRST PARAMETER MUST BE EVENT 06210130
BEGIN 06210140
IF STEP1 ≠ LEFTPAREN THEN BEGIN ERR(105); GO EXIT END; 06210150
IF STEP1 ≠ EVENTID THEN BEGIN ERR(554); GO EXIT END; 06210160
T2=ELBAT[I]; % THE EVENT 06210170
IF REAL(IPCJOB), [33:15] ≠ 0 THEN % COMPILING AN INVOKED IPC JOB 06210180
IF T2, [9:7] = 65 AND T1 ≤ 2 THEN % CANT FIX OR FREE A 06210190
FLAG(555); % FORMAL LEVEL 1 EVENT 06210200
STEPIT; 06210210
END; 06210220
GO TO FUNCT1+1; % TO PROPER SYNTAXER 06210230
OTHERS; GO EXIT; % ACTUAL BOOLEAN INTRINSICS WOULD GO HERE 06210240
FIX; % MAKE THE EVENT ***NOT*** AVAILABLE AND RETURN FORMER STATE 06210250
FREE; % MAKE THE EVENT **** AVAILABLE **** AND RETURN FORMER STATE 06210260
EMITV(T2); EMITO(DUP); % 2 COPIES ON STACK 06210270
EMITPAIR(HAPBIT, LND); % RESETS EVERYTHING BUT HAPBIT 06210280
IF T1=1 THEN EMITPAIR(AVLBIT, LOR); % FIX = BIT ON SO NOT AVAIL 06210290
EMITN(T2); EMITO(STD); % CHANGE IT 06210300
EMITO(LNG); % RETURN TRUE IF IT WAS AVAILABLE, 06210310
GO CKRPN; % 06210320
AVAILABLE; % RETURN TRUE IF THE EVENT IS AVAILABLE (AVL BIT OFF) 06210330
EMITV(T2); EMITO(LNG); GO CKRPN; 06210340
ACQUIRE; % WAIT FOR AVLBIT OR TIME OUT 06210350
PEND; % WAIT TILL HAPBIT OR TIME OUT 06210360
EMITN(T2); % ADDRESS OF THE EVENT 06210370
EMITL(IF T1=4 THEN AVLBIT ELSE HAPBIT); 06210380
IF ELCLASS≠COMMA THEN BEGIN ERR(556); GO EXIT END; 06210390
STEPIT; AEXP; % TIME OUT IN SECONDS 06210400
EMITPAIR(44, COM); % GO DO IT 06210410
EMITO(BEL); EMITO(DEL); % LEAVE RESULT OF FUNCTION ON STACK 06210420
GO CKRPN; 06210430
HAPPENED; % RETURNS TRUE IF THE EVENT HAS THE HAPPENED BIT ON 06210440
EMITV(T2); EMITPAIR(HAPBIT, LND); EMITPAIR(HAPBIT, EQL); 06210450
GO CKRPN; % 06210460
NOBITSON; % NOBITSON(X) = NOT X = NOT 0 06210470
PANAS; EMITO(LNG); EMITPAIR(0, LNG); EMITO(EQL); 06210480
GO EXIT; 06210490
CKRPN; IF ELCLASS ≠ RTPAREN THEN ERR(104); ELSE STEPIT; 06210500
EXIT; 06210510
END BOOLEAN IMPLIED FUNCTIONS; 06210520
IF ELCLASS=TASKID THEN GO TO L33; % ALWAYS ARITHMETIC 06210530
GO TO L18; % FOR SAFETY ONLY 06210540
L11; IF BOOLEAN(ELBAT[I], V0) THEN % BOOLEAN INTRINSIC 06210550
BEGIN BIMPFUN; GO TO TD END; % ALLOW , AND & AFTER IT 06227550
L18; L19; L20; L22; L23; L24; L26; L27; L28; L31; L32; L33; 06234500
06235000
06235500
06236000

```

```

IF ACLASS=TASKID THEN GO TO LODPOINT; % NAME CALL ON TASKID 07217515
IF ACLASS=EVENTID THEN GO TO L22; % NAME OR VALUE CALL EVENTID 07217517
COMMENT IF THE INTRINSIC IS ONLY IMPLIMENTED IN LINE EMIT AN ERR; 07315100
IF TAKE(WHOLE+WHOLE,[33:2]) < 0 THEN FLAG(552); 07315200
IF ELCLASS=TASKID THEN 07529300
    BEGIN TASKATTRIBUTEHANDLER(FS); GO EXIT END; 07729400
PROCEDURE IPCSTATEMENT(W); VALUE W; INTEGER W; 07560000
    BEGIN % VALUE OF W PASSED 1 2 3 4 5 07760100
    LABEL EXIT; LIBERATE,PROCURE,WAIT,CAUSE,RESET; 07560200
    SWITCH STMT:= LIBERATE,PROCURE,WAIT,CAUSE,RESET; 07560300
    DEFINE HAPBIT=8; % BIT [44:1] IS ON IF THE EVENT HAS HAPPENED 07560400
    AVLBIT=1; % BIT [47:1] IS ON IF THEN EVENT IS NOT AVAILIABLE 07560500
    REAL E; 07760600
    IF STEPI # LEFTPAREN THEN BEGIN ERR(105); GO EXIT END; 07760700
    IF STEPI # EVENTID THEN BEGIN ERR(557); GO EXIT END; 07760800
    E:=ELBAT[I]; 07560900
    GO TO STMT[W]; 07561000
    LIBERATE! % CALL COM 44 PASSING! EVENT ADDR, 0, NOT 0 = RESETS AVLBIT 07561100
    PROCURE! % CALL COM 44 PASSING! EVENT ADDR, 1, NOT 0 = GETS CONTROL 07561200
    WAIT! % CALL COM 44 PASSING! EVENT ADDR, 8, NOT 0 = SLEEPS TILL HAP 07561300
    EMITN(E); 07561400
    EMITL(IF W=1 THEN 0 ELSE IF W=2 THEN AVLBIT ELSE HAPBIT); 07561500
    EMITL(0); IF W=1 THEN EMIT0(LNG); EMITPAIR(44,COM); 07561600
    EMIT0(DEL); EMIT0(DEL); EMIT0(DEL); % CUT BACK STACK 07561700
    GO EXIT; 07561800
    CAUSE! % TURN ON HAPPENED BIT 07561900
    RESET! % TURN OFF HAPPENED BIT 07562000
    EMITV(E); 07562100
    EMITPAIR(HAPBIT,IF W=4 THEN LOR ELSE LNG); 07562200
    IF W=5 THEN EMIT0(LND); 07762300
    EMITN(E); EMIT0(STD); 07562400
EXIT! IF STEPI # RTPAREN THEN ERR(104) ELSE STEPIT; 07562500
    END IPC STATEMENT; 07562600
PROCEDURE PROCESSTATEMENT; 07565000
    BEGIN % HANDLES! PROCESS IPCEXTERNAL(AP1,AP2,AP3)[TASKID]; 07565025
    REAL E,T; 07765050
    INTEGER N,MARK,PCNT; 07765075
    LABEL BAD,Z; 07565100
    DEFINE FAULT(F) = BEGIN TI=F; GO BAD END#, FF=18:15; 07565125
    IPCJOB,[FF]=BOOLEAN(REAL(IPCJOB,[FF])+1); % COUNT PROCESS STMENTS 07565150
    IF STEPI # IPCPROCID THEN FAULT(559); % MUST BE EXTERNAL PROCEDURE 07765175
    SCATTERELBAT; 07565200
    EMIT0(MKS); % MARK STACK FOR CHAR MODE CUT BACK WITH XITR AFTER COM 07565250
    IF NT=TAKE(MARK=LINKF+INCRF)=0 THEN GO TO Z; % % PARAMETERS 07565275
    IF STEPI # LEFTPAREN THEN FAULT(105); 07765300
    DO BEGIN 07765325
        IF BOOLEAN((TI=TAKE(MARK=PCNT+1),VO) THEN % VALUE 07565350
            BEGIN STEPIT; % 07565375
                IF EXPRSS=(IF T,CLASS=BOOID THEN BTYPE ELSE ATYPE) THEN 07765380
                    FAULT(123); T:=3; 07765385
            END % 07765390
            ELSE % 07765400
                BEGIN % NAME CALL 07565425
                    IF STEPI # T,CLASS THEN FAULT(123); E:=ELBAT[I]; 07565450
                    IF BOOARRAYID $ ELCLASS $ INTARRAYID THEN 07565475
                        BEGIN EMITPAIR(E,LOD); T:=4; 07765500
                            DO UNTIL STEPI=COMMA OR ELCLASS=RTPAREN; 07765525

```

```

END
ELSE
  BEGIN EMITN(E);
    T1=IF ELCLASS=EVENTID THEN 1 ELSE 2; STEPIT;
  END;
  END NAME CALLS;
  EMITL(T); % PARAMETER TYPE
END
UNTIL ELCLASS # COMMA OR PCNT=N;
IF PCNT # N THEN FAULT(128);
IF ELCLASS # RTPAREN THEN FAULT(104);
Z1 IF STEPI # LFTBRKET THEN FAULT(560);
EMITL(N); % NUMBER OF PARAMETERS
IF STEPI # TASKID THEN FAULT(560);
EMITPAIR(ELBAT(I),LOD);
IF STEPI # RTBRKET THEN FAULT(560) ELSE STEPIT;
EMITPAIR(42,COM); % GO DO IT
EMITN(XITR); % CALLS FIRST SYL OF SEG 1 =EXC= TO CUT BACK STACK
IF FALSE THEN
  BADI ERRCT);
  END PROCESS STATEMENT;
  PROCEDURE TASKATTRIBUTEHANDLER(F); VALUE F; INTEGER F;
  COMMENT TASK ATTRIBUTE HANDLER PROCESSES ALL TASK ATTRIBUTES,
  IT IS EXPLICITLY CALLED IN 2 PLACES:
  1: IN $TMT PASSING F = FS,
  THIS HANDLES TASK ATTRIBUTE ASSIGNMENTS BY CALLING AEXP,
  2: IN PRIMARY PASSING F = FP,
  THIS ALWAYS LEAVES A VALUE ON THE STACK, IF AN ASSIGNMENT
  IS REQUIRED IT IS MADE WITH A "SND". IF NOT THE CURRENT
  VALUE OF THE ATTRIBUTE IS LEFT ON THE STACK,
  ACTUALPARAMETERPART AND BPRIMARY DIRECT TRAFFIC THRU THE
  ABOVE ENTRY POINTS TO DO THEIR WORK,
END COMMENT;
BEGIN
  REAL E; INTEGER ATNDX;
  E:=ELBAT(I); % TASK ID
  IF STEPI # PERIOD THEN
    BEGIN
      STEPIT;
      IF ATNDX#
        IF Q = "9TASKV" THEN 0 ELSE
        IF Q = "4MFID0" THEN 1 ELSE
        IF Q = "3FID00" THEN 2 ELSE
        IF Q = "6STATU" THEN 3 ELSE
        IF Q = "7STACK" THEN 4 ELSE 562
      =562
      THEN FLAG(562); % UNRECOGNIZED ATTRIBUTE
      IF STEPI # ASSIGNOP THEN
        BEGIN % SET ONE
          IF ATNDX#4 THEN FLAG(563); % CANT BE CHANGED
          STEPIT; AEXP;
          IF ATNDX#3 THEN % TSK, STATUS REQUIRES A COM FOR POSSIBLE DS,
            BEGIN EMITPAIR(E,LOD); EMITPAIR(45,COM);
              EMIT(DEL); IF F#FP THEN EMIT(DEL); % CUT BACK STK
            END %
          ELSE % JUST CHANGE THE ATTRIBUTE
            BEGIN EMITL(ATNDX); EMITN(E);

```

```

07765550
07565575
07765600
07765610
07765615
07765625
07565650
07765675
07565700
07565725
07765750
07765775
07565800
07765825
07565850
07765875
07765900
07565925
07565950
07765975
07765985
07566000
07766025
07766050
07766075
07566100
07766125
07566150
07766175
07566200
07566225
07766250
07766275
07766300
07566325
07566350
07766375
07766400
07566425
07566450
07566475
07566500
07566525
07566550
07566575
07766600
07566625
07566650
07566675
07766700
07566725
07766730
07566735
07766740
07566745
07766750
07566755

```

```

EMIT0(IF F=FP THEN SND ELSE STD);
END;X
END
ELSE
IF F=FS THEN FLAG(564) % MISSING I=
ELSE
BEGIN EMITL(ATNDX); EMITV(E) END;
END
ELSE ERR(561); % MISSING ".
END TASK ATTRIBUTE HANDLER;
NI= IF Q = "8LIBER" THEN 1 ELSE
IF Q = "7PROCU" THEN 2 ELSE
IF Q = "5CAUSE" THEN 4 ELSE
IF Q = "5RESET" THEN 5 ELSE 0;
IF N # 0 THEN BEGIN IPCSTATEMENT(N); GO XXX END;
IF Q = "7PROCE" THEN BEGIN PROCESSTATEMENT; GO XXX END;
BEGIN % IF I+2 IS EVENTID THEN ITS IPC WAIT
IF TABLE(I+2)=EVENTID THEN
BEGIN IPCSTATEMENT(3); GO XXX END;
NI=1; CI=2;
END ELSE %
OCT0132000000140000, "3FIX00",
OCT01300000006240000, "4NABS0",
OCT01320000000240000, "4FREE0",
OCT01320000000640000, "4PEND0",
OCT01300000000040000, "3TAN00",
OCT01300000006360000, "8SYSTE", "MID00000",
OCT01320000000760000, "8NOBIT", "SON00000",
OCT01320000000460000, "7ACQUI", "RE000000",
OCT01320000000360000, "9AVAIL", "ABLE0000",
OCT01320000000560000, "8HAPPE", "NED00000",
OCT04300000260000000, "4TASK0",
OCT04300000270000000, "5EVENT",
OCT14400000250000000, "4SIZE0",
OCT00300000260000000, "4FAST0",
OCT00300000270000000, "4SLOW0",
OCT20000000000004050, COMMENT POWERS OF TEN ;
" " ; COMMENT SORTA ;
" " ; COMMENT LASTSEQUENCE, LASTSEQROW ;
692 STEP 3 UNTIL 728, 732 STEP 4 UNTIL 748, 750 STEP 2 UNTIL 756
DO UNTIL STEPI=BEGINV OR %
ELCLASS=DECLARATORS AND ELBAT[I], ADDRESS=PROCV; % IPC
IF ELCLASS#BEGINV THEN % COMPILE AN INVOKABLE IPC PROGRAM,
BEGIN
IPCJOB:=TRUE; I:=I-1; % BACK UP TO LOOK AT PROCEDURE
END;
IF REAL(IPCJOB) # 0 THEN % INVOKED OR INVOKING IPC
BEGIN
IF MARK:=REAL(IPCJOB), [33:15] DIV 2 # 0 THEN % INVOKED IPC
IF IDARRAY[8]:=NI=TAKE(MARK) # 0 THEN % HAS PARAMETERS
BEGIN % BUILD PARAMETER DESCRIPTION SEGMENT
LI=0;
DO PRTIO, LJI= %
IF ELCLASS:=(TI=TAKE(MARK+LI=L+1)), CLASS
= EVENTID THEN 1 ELSE
IF BOOARRAYIDSELCLASS$INTARRAYID THEN 4 ELSE
IF BOOLEAN(T,VO) THEN 3 ELSE 2

```

```

07766760
07566775
07766800
07766825
07766850
07766875
07566900
07766925
05766950
07566975
07810200
07810210
07810220
07810230
07810240
07810250
07825000
07825100
07825200
07825300
07825400
871309214421
871609214422
871909214423
872209214424
872509214425
872809214426
873209214427
873609214428
874009214429
874409214430
874809214431
875009214432
875209214433
8754 09214435
8756 09214436
8758 09214437
8759 09214440
8760 09214445
09214513
09252000
09252025
09252100
09252200
09252300
09252400
09405500
09405505
09405510
09405515
09405520
09405525
09405530
09405535
09405540
09405545
09405550

```

Moore Business Forms, Inc. sv 14171

```

UNTIL L=N)
IDARRAY[9]=MOVEANDBLOCK(PRT,N+1)
END INVOKED IPC JOB
IDARRAY[2],[3]=2+REAL(MARK#0)
COMMENT SEG0[2],[3]=JAR[2],[6]=IPC JOB
SEG0[2],[4]=JAR[2],[7]=INVOKED IPC JOB
END IPC SET UP
IF BOOARRAYIDSTYPESINTARRAYID OR TYPE=TASKID THEN
IF (FORMALF=PTOG) AND NOT P2 THEN
BEGIN
PJI=PJ+1) % COUNT PARAMETERS
ADDRSFI=IF IPCJOB THEN GETSPACE(TRUE,1)+64*PJ % PRT
ELSE PJ % STACK
END
ELSE
IF STOPGSP THEN ADDRSFI=0
ELSE ADDRSFI=GETSPACE(P2,1)
IF BOOARRAYIDSTYPESINTARRAYID OR TYPE=TASKID AND NOT IPCJOB
GT1=IF PTOG THEN IF IPCJOB THEN LEVEL ELSE LEVELF ELSE LEVEL
$ VOID 13371001
PROCEDURE TASKS % DOES TASK DECLARATION
BEGIN
LABEL EXIT)

JUMPCHKX) %
IF GTA1[J]=1) #0 THEN FLAG(25) % CANT USE OWN OR SAVE

TYPEV=TASKID)
IF NOT SPECTOG THEN EMIT0(MKS)
ENTRY(TYPEV) % DOES ALL OF THE LIST
IF ELCLASS#SEMICOLON THEN FLAG(0)
IF SPECTOG THEN GO EXIT)
EMITL(5) EMITL(1) EMITL(GTA1[0]) % SIZE, # DIMENSIONS, # TASK
EMITPAIR(1,SSN) % ASK BLOCK CONTROL FOR SAVE TASK ARRAYS
EMITV(5) % CALL BLOCK CONTROL
EXIT)
END TASK ENTR)
IPCOMPLETE,TASKDEC,EVENTDEC, % I P C
AUXSIZERR, % BAD USE OF AUXMEM OR SIZE
PTRDEC,DEFINDEC,AUXSIZERR,AUXSIZERR,TASKDEC,EVENTDEC)
LPB=[36112]#, PJFLD=[1615]#, %
$ VOID 14125001
IF SPECTOG THEN % DOING A PROCEDURE SPECIFICATION
IF BUP=PJ THEN % ALL SPECIFICATIONS PROCESSED, DO CLEAN UP,
BEGIN
F1=FZER0) % 3*1600 = F+0, PARAMETER BASE ADDRESS
DO BEGIN % LOOK THRU PARAMS LAST TO FIRST & PUT IN PROC+INFO
G1=TAKE(BUP=LASTINFO) % ELBAT OF LAST PARAM SPECIFIED
WHILE PJ#(IF IPCJOB THEN G,PJFLD ELSE G,ADDRESS)
DO G1=TAKE(BUP=BUP-TAKE(BUP+1),PURPT) % GET PJ=TH ONE
TYPEV=G,CLASS)
IF IPCJOB THEN
G,PJFLD=0 %
ELSE
IF TYPEV=FRMTID OR TYPEV=SUPERFRMTID THEN
G,ADDRESS=F1=F+2 % LEAVE ONE FOR INDEX
ELSE
IF BOOARRAYID ≤ TYPEV ≤ INTARRAYID THEN

```

```

09405555
09405560
09405565
09405570
09405575
09405580
09405585
13330550
13342000
13342250
13342500
13342750
13343000
13343500
13344000
13344500
13345000
13346000
13170000
13371000
13595025
13595050
13595075
13595100
13595125
13595150
13595175
13595200
13595225
13595250
13595275
13595300
13595325
13595350
13595375
13595400
13595425
14016700
14016800
14021000
14023000
14069999
14070000
14071000
14072000
14073000
14074000
14075000
14076000
14077000
14078000
14079000
14081000
14082000
14083000
14084000
14085000
14086000

```

14121
 14122
 14123
 14124
 14125
 14126
 14127
 14128
 14129
 14130
 14131
 14132
 14133
 14134
 14135
 14136
 14137
 14138
 14139
 14140
 14141
 14142
 14143
 14144
 14145
 14146
 14147
 14148
 14149
 14150
 14151
 14152
 14153
 14154
 14155
 14156
 14157
 14158
 14159
 14160
 14161
 14162
 14163
 14164
 14165
 14166
 14167
 14168
 14169
 14170
 14171
 14172
 14173
 14174
 14175
 14176
 14177
 14178
 14179
 14180
 14181
 14182
 14183
 14184
 14185
 14186
 14187
 14188
 14189
 14190
 14191
 14192
 14193
 14194
 14195
 14196
 14197
 14198
 14199
 14200

```

BEGIN 14087000
  GT1:=N1=TAKE(BUP+T1:=G,INCR); 14088000
  G,ADDRESS:=F1:=F+N1:=N+1; %LEAVE ROOM FOR LOWER BOUND 14089000
  WHILE N1:=N-1#0 DO 14090000
    IF T2:=TAKE(BUP+T1+N) < 0 THEN 14091000
      PUT(ABS(T2&(4*(F=N)+2)LPB),BUP+T1+N) 14092000
    END 14093000
  ELSE G,ADDRESS:=F1:=F+1; 14094000
  PUT(G,BUP); G,INCR:=GT1; % UPDATE PARAMETER INFO 14095000
  IF FWDTOG THEN 14096000
    BEGIN 14097000
      IF (GT1:=TAKE(MARK+PJ)),CLASS#G,CLASS THEN 14098000
        FLAG(49); % CLASS ERROR FORWARD TO ACTUAL 14099000
      IF GT1,VO#G,VO THEN FLAG(50); 14100000
    END 14101000
  ELSE 14102000
    PUT(G,MARK+PJ); % PROCEDURE ADDITIONAL INFO ENTRY 14103000
  END 14104000
  UNTIL PJ:=PJ-1=0; % 14105000
  SPECTOG:=FALSE; 14106000
  IF IPCJOB THEN % WRAP UP IPC SET UP & START COMPILE 14107000
    BEGIN 14108000
      IPCJOB:=BOOLEAN(2*MARK); % FALSE = POINTS TO + INFO 14113000
      IF STEP1#BEGINV THEN FLAG(551); 14114000
      ELBAT[1],CLASS:=SEMICOLON; 14115000
      GO TO START; % NOW WE CAN COMPILE THE IPC JOB 14116000
    END; 14117000
    GO TO HF; % NOT IPC, GO COMPILE THE PROCEDURE 14118000
  END SPECIFICATION WRAP UP; 14119000
  STOPDEFINE:=DUMPV#(GTA1[J1:=J+1]:=ELBAT[1],ADDRESS)#MONITORV; 14129000
  ERRORTOG:=TRUE; 14130000
  IF IPCJOB THEN % 14133100
    IF SPECTOG THEN % VERIFY THAT PROPER TYPES ARE BEING PASSED 14133200
      IF BOOV <= G1:=GTA1[J] <= INTV OR G=ARRAYV OR G=EVENTV THEN % OK 14133300
        ELSE FLAG(550); % 14133400
    AUXSIZERR; % 14135500
    TASKDECI TASK; GO TO GOTSCHK; 14155500
    EVENTDECI P1:=TRUE; ENTER(EVENTID); GO START; 14155700
    SCATTERELBAT; 14302100
    IF IPCJOB THEN % MAKE "MYSELF" TASK ARRAY ENTRY AT R+@25 14302200
      BEGIN % 14302300
        TYPEV:=TASKID; COUNT:=6; SCRAM:= (ACCUM[1]#"6MYSEL") MOD 125; 14302400
        ACCUM[2]#"08"F#[114315]; 14302500
        GO DOITASYWAY; 14302600
      END; % 14302700
      % VOID 14305001 14305000
    IF NOT IPCJOB THEN 14323000
      IF LEVEL < 31 THEN LEVMODE[LEVEL:=LEVEL+1]:=MODE ELSE FLAG(39); 14323100
      PJ:=0; 14323200
      % VOID 14324001 14324000
    IF STEP1#SECRET THEN FLAG(010) % WAS NOT IN PARAMETER LIST 14345000
    ELSE 14346000
      BEGIN 14347000
        G:=ELBAT[1]; 14348000
        IF 1$(IF IPCJOB THEN G,PJFLD ELSE G,ADDRESS)SPJ THEN 14349000
          ELSE FLAG(10); % WAS NOT IN FORMAL PARAMETER LIST 14350000
        PUT(TAKE(G)&1[1014711],G); % MARK AS VALUE PARAMETER 14351000
      END

```

	END *	14352000
\$	VOID 14354001	14353000
	IF IPCJOB AND PJ=0 THEN * 0 PARAM IPC PROCEDURE DEC	14377000
	GO TO IPCOMLETE) *	14378000
	IF SPECTOG THEN GO TO START) * AND PROCESS THE SPECIFICATIONS	14379000
	IF STEPI=FORWARDV OR Q="8EXTER" * EXT ILLEGAL PARAM CK IN PROCESS	14467000
	IF ELCLASS # FORWARDV THEN * EXTERNAL	14469025
	BEGIN	14469050
	IF PROINFO.CLASS # PROCID THEN FLAG(558) * MUST BE UNTYPE	14469075
	ELSE PUT(TAKE(PROINFO)&IPCPROCID CLASS,PROINFO);	14469100
	END *	14469125
	ELSE *	14469150
\$	PATCH NUMBER 411 FOR XALGOL CONTAINS 22 CARDS	
\$	THIS DOCUMENTS THE IPC ERROR MESSAGES, 1/5/73	
021	BLOCK1 SAVE OR AUXMEM APPEARS IMMEDIATELY	00041000
	BEFORE IDENTIFIER (NO TYPE),	00042000
550	IPCROGEDUREDEC1 AN IDENTIFIER OF THIS CLASS CAN NOT BE	00306900
	PASSED TO AN INVOKED PROGRAM,	00306905
551	IPCROGEDUREDEC1 MISSING BEGIN AFTER SPECIFICATION PART,	00306910
552	ACTUALPARAPART1 THIS INTRINSIC CAN NOT BE PASSED AS A	00306915
	FORMAL PARAMETER,	00306920
553	PROCESSTATEMENT1 A TASK IDENTIFIER MAY NOT BE PASSED AS AN	00306925
	EXPLICIT PARAMETER TO AN INVOKED PROCEDURE,	00306930
554	BIMPFUN1 PARAMETER MUST BE AN EVENT,	00306935
555	BIMPFUN1 AN INVOKED PROCESS CAN NOT FIX OR FREE AN	00306940
	EVENT PASSED TO IT,	00306945
556	BIMPFUN1 MISSING COMMA IN ACQUIRE OR PEND FUNCTION,	00306950
557	IPCSTATEMENT1 PARAMETER MUST BE AN EVENT,	00306955
558	BLOCK1 AN EXTERNAL IPC PROCEDURE MUST BE UNTYPED,	00306960
559	PROCESSTATEMENT1 PROCEDURE MUST BE DECLARED EXTERNAL,	00306965
560	PROCESSTATEMENT1 MISSING OR INCORRECT TASK PART,	00306970
561	TASKATTRIBUTEHANDLER1 MISSING , IN TASK ATTRIBUTE PART,	00306975
562	TASKATTRIBUTEHANDLER1 MISSING OR UNDEFINED TASK ATTRIBUTE,	00306980
563	TASKATTRIBUTEHANDLER1 TASK ATTRIBUTE IS NON ASSIGNABLE,	00306985
564	TASKATTRIBUTEHANDLER1 MISSING ASSIGNMENT OPERATOR IN TASK	00306990
	ATTRIBUTE ASSIGNMENT STATEMENT,	00306995
\$	PATCH NUMBER 412 FOR XALGOL CONTAINS 8 CARDS	
\$	THIS PROVIDES THE "RUN" STATEMENT FOR IPC, ONLY VALUE PARAMETERS	
\$	CAN BE PASSED TO PROCESSES THAT IS "RUN", 3/19/73,	
565	PROCESSTATEMENT1 PARAMETERS TO A "RUN" TASK MUST BE VALUE,	00307000
	* AND * RUN IPCVALUEONLYPROCEDURE(API,AP2)[TASKID];	07765035
	REAL E,T) BOOLEAN RUNNER) * SET TRUE IF CALLED AS A "RUN"	07565050
	RUNNER:=Q,[4216]=0) * 5TH CHARACTER OF "RUN" = 0, PROCESS = E,	07565135
	IF RUNNER THEN FAULT(565); * ALL RUN PARAMS MUST BE VALUE	07765435
	IF RUNNER THEN EMIT0(SSN); * STARTUSERTASK USES = FOR A "RUN"	07765815
	IF Q="7PROCE" OR Q="3RUN00" THEN * IPC INITIATION	07810245
	BEGIN PROCESSTATEMENT1 GO XXX END) *	07810250
\$	PATCH NUMBER 413 FOR XALGOL CONTAINS 2 CARDS	
\$	MAKES TEMPORARY ASSIGNMENT FOR EXTERNAL PROCEDURE DECLARATIONS,	
\$	3/20/73	
	SPRT[(PRTIMAX1=PRTIMAX-1),[3815]]:= * REUSE PRT CELL	14469110
	BOOLEAN(*) AND NOT BOOLEAN(2*PRTIMAX,[4315]);	14469115
\$	PATCH NUMBER 414 FOR XALGOL CONTAINS 1 CARD	
\$	STOPS DEFINES WHIN PICKING UP ATTRIBUTE NAME, 3/22/73	
	STOPDEFINE1=TRUE) STEPIT) * PICK UP ATTRIBUTE NAME = NO DEFINES	07566425
\$	PATCH NUMBER 415 FOR XALGOL CONTAINS 6 CARDS	
\$	THIS CHANGE INSURES THAT THE LEXICOGRAPHICAL LEVEL OF THE TASK ID	00000010


```

S: IN A PROCESS STATEMENT IS NO LOWER THAN THE "DEEPEST" NAME 00000020
S: PARAMETER IN AN INVOKATION, 8/4/73 DHB 00000030
566 PROCESSTATEMENT: LEXICAL LEVEL OF TASK LESS THAN A NAME PARAM, 00307005
    INTEGER N, MARK, PCNT, MAXLVL; % 07765075
        IF STEPI * T, CLASS THEN FAULT(123); % WRONG TYPE 07565450
        IF (E:=ELBAT[I]), LVL > MAXLVL THEN MAXLVL:=E, LVL; % 07765465
        IF (E:=ELBAT[I]), LVL < MAXLVL AND MAXLVL > 0 THEN FAULT(556); % 07765835
    EMITPAIR(E, LOD); % 07765850
S: PATCH NUMBER 416 FOR XALGOL CONTAINS 4 CARDS
S: THIS CHANGE INSURES THAT AN IPC PROCEDURE IS NOT COMPILED TO "GO" AND 00000010
S: CHECKS TO SEE THAT NO MORE THAN 29 PARAMETERS ARE ALLOWED IN AN 00000020
S: IPC PROCEDURE DECLARATION, (SEE MCP PATCH 708) 8/4/73 DHB 00000030
567 PROGRAMI COMPIL AND GO OF INVOKED IPC PROCEDURE NOT ALLOWED, 00307010
568 BLOCK: AN INVOKED IPC PROCEDURE CAN HAVE ONLY 29 PARAMETERS, 00307015
    IF SAVETIME=0 THEN BEGIN ERRORTOGI:=TRUE; FLAG(567) END; %GO WHERE 09252250
    IF IPCJOB THEN IF PJ > 29 THEN FLAG(568); % TOO MANY PARAMETERS 14335000
S: PATCH NUMBER 417 FOR XALGOL CONTAINS 5 CARDS
S: THIS ADDS A NEW TASK ATTRIBUTE "INITIATOR" WHICH CAN BE SET AND READ, 00000010
S: IF "INITIATOR" IS NON ZERO WHEN A TASK IS INITIATED IT WILL APPEAR AS 00000020
S: IF IT ORIGINATED FROM THE STATION WITH THE INDICATED INTERNAL LINE 00000030
S: NUMBER, 11/21/73 DHB 00000040
    IF Q = "7STACK" THEN 4 ELSE 07566575
    IF Q = "9INITI" THEN 5 ELSE 07566580
    562 % 07766599
    = 562 % 07766600
    EMITL(6); EMITL(1); EMITL(GTA1[0]); % SIZE, # DIMISIONS, # TASK 13595325
S: PATCH NUMBER 418 FOR XALGOL CONTAINS 2 CARDS
S: THIS PROVIDES AN INDEXED ARRAY ROW WRITE OF THE FORM: 00000010
S: WRITE(FIL, AEXP, ARRY[INDX]); 00000020
S: IT IS THE USERS RESPONSIBILITY TO INSURE THAT AEXP+INDX DOES NOT 00000030
S: EXCEED THE ARRAY BOUNDS, 11/24/72 00000040
    "410" % LOCAL PATCH LEVEL 11/24/72 01831100
S VOID 08672001 08668000
S: PATCH NUMBER 419 FOR XALGOL CONTAINS 5 CARDS
S: THIS PROVIDES THE CAPABILITY TO PROGRAMATICALLY GENERATE A MEMORY
S: DUMP IN ALGOL OR XALGOL BY THE STATEMENT: -1/15/73
S: DUMPNOW(INTEGER);
    "411" % LOCAL PATCH LEVEL 1/15/73 DHB 01831100
    IF Q="7DUMPN" THEN % DUMPNOW(W) MEMORY DUMP 07810500
    BEGIN PAN; EMITL (0); EMITPAIR(48, COM); % 07810525
    EMIT0(DEL); EMIT0(DEL); GO TO XXX; % 07810550
    END; % 07810575
S: PATCH NUMBER 420 FOR XALGOL, XV, 1 CONTAINS 62 CARDS
S: THIS MAKES EFFICIENCY IMPROVEMENTS TO THE XREF AND BEND PATCH 7/29/73 00000010
S VOIDT 01001778 01001750
ARRAY XREFAY1[0:10], XREFAY2[0:29], XINFO[0:31, 0:127]; % FOR XREF ONLY 01002000
INTEGER XREFPT, XLUN; % 01002100
DEFINE XREFINFO[R, C]=XINFO[R, (C) DIV 2]; % 01002200
XMARK:=IF XREF THEN XREFAY2[XREFPT-1]:=NABS(*)#; % MARK ASSIGNMENT 01002300
ARRAY BEGINSTACK[0:31]; INTEGER BSPOINT; % FOR BEND 01002400
BOOLEAN DEFINING; % BEND 01002500
S VOIDT 0100765; 01007100
    STREAM PROCEDURE STUFF(LVL, N, L); VALUE LVL, N; 02196530
        DII=L; DSI=8 LIT " "; SII=L; DSI:=13 WDS; DII=DI=4; 02196550
        DSI=L; IT[""]; SII=LOC LVL; DSI=2 DEC; DSI=LIT[""]; DSI=8 DEC; 02196560
    STUFF(BSPOINT, BEGINSTACK[BSPOINT], LIN); 02196580
    IF Q = "4XREF0" THEN BEGIN SWITCHIT(XREFBIT); GO AGAIN END; 02419100

```



```

$ VOID 02419121                                02419110
IF NOT ENDTG THEN % DONT XREF COMMENTS FOLLOWING END 02882050
IF T.[35113] > SORTA THEN % NOT "ALGOL" IDENTIFIER 02882200
PTOGI=TRUE; % DONT XREF THESE THINGS 05442500
PTOGI=FALSE; % 05443500
IF XREF THEN % DONT COUNT THIS OCCURANCE IF 10264802
XREFPT=XREFPT-REAL(ELBAT[1],LVL#0);%SEEN B4 10264804
IF XREF THEN % DONT XREF SECOND DEC OF SAME VAR 13348000
XREFPT=XREFPT-REAL(LEVEL#0); % GET RID OF LAST XREF 13348100
END; % 13348200
DEFININGI=TRUE; % WE ARE 14255500
IF XREF THEN % DONT COUNT THIS AS AN OCCURANCE 14259077
XREFPT=XREFPT-REAL(ELBAT[1],LVL#0); % IF SEEN B4 14259078
PTOGI=TRUE; E; PTOGI=FALSE; % DONT XREF DEFINE PARAMS 14259096
DEFININGI=FALSE; % WE ARE NOT 14268500
IF XREF THEN % FORGET THE PARAMETERS SPECIFED IN A FORWARD 14470900
IF GI=TAKE(MARK) > 0 THEN % OR EXTERNAL DECLARATION 14470910
BEGIN SPACE(DSK1,-G); XLUN=XLUN-G END; 14470920
IF XLUN > 0 AND ERRORCOUNT = 0 THEN % WE DID SOME XREFFING SO DOITTOIT 17001000
IF NOT NOHEADING THEN WRITE(LINE[PAGE]); DATIME; 17002520
$ VOID 17055559
STREAM PROCEDURE PUP(S,D); 17047001
BEGIN 17048000
SI*S; DI=D; 17049000
60(IF SC=" " THEN JUMP OUT; DS=CHR); 17050000
DS=10LIT" "; SI=S; SI=SI+32; SI=SI+28; % TO SEG NUM 17051000
DS=4CHR; DS=3LIT" *"; DS=8CHR; DS=LIT"*"; 17052000
END; 17053000
DEFINE BLANK(P) = BLANKET(14,P);# 17054000
PAY=LIN#; 17063000
$ VOID 17069999 17064000
DO BEGIN 17065000
IF XREFPT=XREFPT+1=30 THEN 17073500
BEGIN 17074000
READ(DSK2,30,XREFAY2[*])(EOF); 17075000
XREFPT=0; 17076000
END; 17077000
IF (I=XREFAY2(XREFPT)), [21:27]>99999999 THEN GO TO EOF; 17078000
END 17079000
UNTIL I,[9:12] GTR 0; % DONT SORT NON XREFED ENTRIES 17079250
BLANK(PAY); 17079500
IF LASTADDRESS NEQ LASTADDRESS+A[0] THEN % NEW OCCURANCE 17083000
IF LASTADDRESS,[9:12] GTR XREFAY1[9] THEN 17094100
BLANK(PAY); XREFPT=0; 17095100
$ VOID 17100101 17099000
BLANK(PAY); 17100000
PUP2(LASTADDRESS LSS 0,LASTADDRESS,[21:27],XREFPT,PAY[1]); 17101000
$ VOID 17103000 17103000
BLANK(PAY); 17103500
$# PATCH NUMBER 421 FOR XALGOL CONTAINS 287 CARDS 00000020
$! THIS IMPLEMENTS THE B6700 "INCLUDE" $ CARD OPTION AND DE*IMPLEMENTS 00000010
$! THE "CAST" LIBRARY COMPILER FEATURE, IN ADDITION THIS MAKES SEVERAL 00000020
$! IMPROVEMENTS TO THE READACARD ROUTINE, THE HANDELING OF VOID CARDS 00000030
$! IS CORRECTED AND THE "EOF NO LABEL" PROBLEM IN THE COMPILER IS 00000040
$! ELIMINATED, THIS LAST CHANGE MAKES 9"S CARDS OPTIONAL ON BOTH THE 00000050
$! CARD AND TAPE FILES, 00000060
$! NOVEMBER 20, 1973, == DAVE BROWN == U.S. CUSTOMS SAN DIEGO, CA, 00000070
$ VOID 00305081 00305010

```

615	READACARD; UNEXPECTED END OF INPUT,	00415000
620	STARTINCLUDING; NESTED INCLUDES NOT IMPLEMENTED,	00420000
621	STARTINCLUDING; FIRST INCLUDE CARD MUST NAME A FILE,	00421000
622	STARTINCLUDING; ILLEGAL FILE IDENTIFIER,	00422000
623	STARTINCLUDING; ILLEGAL SEQUENCE NUMBER ON INCLUDE CARD,	00423000
624	STARTINCLUDING; FILE MUST HAVE BEEN USED BEFORE FOR * OPTION,	00424000
625	STARTINCLUDING; BAD SYNTAX ON THIS INCLUDE CARD,	00425000
626	STARTINCLUDING; YOUR INCLUDE FILE IS NOT ON DISK,	00426000
627	STARTINCLUDING; BEGINNING SEQUENCE NUMBER IS NOT IN THE FILE,	00427000
	INTEGER ADDVALUE,TOTALNO; %	01000860
	\$ VOIDT 01001521	01001490
	\$ VOIDT	01331000
	THE LAST CHARACTER IN THE TAPE AND CARD BUFFERS, ;	01336000
	\$ VOIDT 01339001	01337000
	0 INCLUDE IN PROCESS,	01394700
	> 3 ELABORATING A DEFINE,	01398000
	\$ VOIDT 01398201	01398100
	SAVE ARRAY DEFINEARRAY[0:25]; % 013*(MAX NEST DEPTH)+1 (SEE 1498100)	01491000
	DEFINE LASTSEQ = DEFINEARRAY[25]; % TEMP STORAGE FOR SEQ OF CARD	01498100
	LSEQM1 = DEFINEARRAY[24]; % FOR CALLING CONV	01498200
	LASTERR = DEFINEARRAY[24]; % HOLDS SEQ NO OF LAST ERROR	01498300
	FILE INCLUDE DISK SERIAL (1,10,150);	01561010
	\$ VOIDT 01561051	01561011
	FILE DSK1 DISK SERIAL [20:300] (2,10,30); % FOR XREF	01561025
	FILE DSK2 DISK SERIAL [20:300] (2,30,30); % FOR XREF	01561050
	SAVE ARRAY CBUFF,TBUFF,IBUFF[0:9]; % INPUT BUFFERS	01561056
	\$ VOIDT 01561451	01561065
	\$ VOIDT 01576001	01575000
	\$ VOIDT 01743301	01742200
	\$ VOIDT 01784001	01756000
	INTEGER SAVLASTUSED,ILCR; % FOR INCLUDE	01783000
	REAL INCLCR,INCLRANGE; % FOR INCLUDE OPTION	01784000
	,#="413" % LOCAL PATCH LEVEL 11/20/73 DHB	01831100
	\$ VOIDT 02001611	02001010
	\$ VOIDT 02013721	02013010
	BOOLEAN PROCEDURE STARTINCLUDING; FORWARD; %	02065700
	BOOLEAN PROCEDURE READINCLUDE; FORWARD; %	02065800
	\$ VOIDT 02134501	02131500
	PROCEDURE WRITELINE; %	02181000
	IF SINGLTOG THEN WRITE(LINE,15,LIN[*]); %	02181250
	ELSE WRITE(LINE[DBL],15,LIN[*]); %	02181500
	\$ VOIDT 02182251	02181750
	DEFINE PRINTCARD = BEGIN %	02182500
	DSI=LIT" "; SII=LOC SYMBOL; SII=SI+7;	02185250
	DSI=CHR; DSI=2LIT" ";	02185500
	SWITCH SWI=LCARD,LCARD,LTAPE; %	02191250
	CARDNUMBER; IF TOTALNO < 0 THEN TOTALNO:=ABS(TOTALNO)*1	02192000
	ELSE TOTALNO:=TOTALNO+ADDVALUE;	02192250
	\$ VOIDT 02193001	02192500
	CHANGESEQ(CARDNUMBER,LCR);	02193250
	IF LASTUSED > 0 OR NEWINCL THEN	02193800
	IF SAVLASTUSED=3 THEN GO TO LTAPE; % INCLUDE CARD WAS ON TAPE	02194600
	\$ VOIDT 02202251	02198500
	DEFINE SEQCOMPARE(TLCR,CLCR) = %	02202500
	\$ VOIDT 02203001	02202750
	MEDIUM="C"; % FROM CARD READER	02203250
	LCRI=TLCR; LASTUSED:=3;	02204000

```

MEDIUMI="T"; % FROM TAPE                                02204250
MEDIUMI="P"; % CARD PATCHES TAPE                       02205500
LASTUSED=3; % IN CASE OF EOF ON TAPE                   02205750
READ(TAPE,10,TBUFF[*])[EOF];                           02206000
$ VOIDT 0220850;                                         02206250
    LASTUSED=2;                                         02209250
    #, % END SEQCOMPARE DEFINE                          02209750
    DIDINCLUDE=REAL(DOLLARTOG)="H"; %                  02210000
    LABEL USETHESWITCH,CARDONLY,CARDLAST,EOF,TAPELAST,COMPAR,EXIT,XIT; 02210250
    BOOLEAN DOLLAR2TOG; %                               02210500
$ VOIDT 02211251;                                       02210750
    IF LASTUSED=1 THEN GO TO CARDONLY; %               02212000
    IF LASTUSED=3 THEN GO TO TAPELAST; %               02212025
    IF LASTUSED=2 THEN GO TO CARDLAST; %               02212050
    IF LASTUSED=0 THEN % INCLUDE                       02212100
    IF READINCLUDE THEN GO USETHESWITCH ELSE GO EXIT; % 02212200
$ VOIDT 02215001;                                       02213500
    LCR=LCR; %                                         02215500
$ VOIDT                                                 02215750
    READ(CARD,10,CBUFF[*])[EOF];                       02216250
    IF LASTUSED=1 THEN GO TO EXIT; % CARD ONLY        02216500
    IF LASTUSED=1 THEN % EOF ON CARD MEANS BYE=BYE    02217250
    BEGIN ERRORTOG:=TRUE; ERR(615) END; % BRANCHES TO ENDOFITALL 02217275
    IF LASTUSED=3 THEN % EOF ON TAPE = SWITCH TO CARD ONLY 02217500
    BEGIN LCR=LCR; LASTUSED=1; MEDIUMI="C"; GO EXIT END; 02217750
    FILL CBUFF[*] WITH 4("% END=OF","=FILE *#"),"COMPILER","99999999"; 02218000
$ VOIDT 0221840;                                       02218250
$ VOIDT 02219001;                                       02218750
    READ(TAPE,10,TBUFF[*])[EOF];                       02219500
$ VOIDT 02224001;                                       02219750
    SEQCOMPARE(TLCR,LCR);                               02224500
$ VOIDT                                                 02224750
$ VOIDT                                                 02225500
    IF COMPARE(MKABS(LASTSEQ),LCR)=1 THEN %            02226000
    PUTSEQNO(LASTSEQ,LCR); %                            02228000
    CARDNUMBER=CONV(LSEQMI,5,8); %                     02228100
    NCR=NCR+32768; % OVER "$" (CHAR MODE POINTER)    02228750
    DOLLARCARD; %                                       02229000
    IF NOT DIDINCLUDE THEN %                           02229200
    IF LISTPTOG OR PRINTDOLLARTOG THEN PRINTCARD;    02229250
    IF DOLLAR2TOG=EXAMIN(FCR)=" " AND EXAMIN(FCR+32768)="$" THEN 02230000
$ VOIDT                                                 02230100
$ VOIDT                                                 02230500
    NCR=NCR+32768+32768; % OVER "$" (CHAR MODE POINTER) 02230750
    IF DIDINCLUDE THEN GO USETHESWITCH; %             02231100
$ VOIDT 02235251;                                       02234251
    OUTPUTSOURCE;                                       02234500
    CARDCOUNT=CARDCOUNT+1;                              02234750
    IF OMITTING OR %                                    02235000
    DOLLAR2TOG THEN GO USETHESWITCH;                  02235250
$ VOIDT                                                 02235500
    OPTIONS(OPINX+1)=REAL(B:=TRUE); %                  02334000
$ VOIDT                                                 02335000
$ VOIDT                                                 02339000
$ VOIDT                                                 02353000
ALONG;                                                 02356500
$ VOIDT                                                 02358000
STREAM PROCEDURE GETVOID(VP,NCR,VR,LCR,SEQ); VALUE NCR,LCR; 02359100

```

```

BEGIN
LABEL L,TRANS;
SII=NCR; DII=VP; DSI=8 LIT "0";
2(34(IF SC="" THEN SII=SI+1 ELSE JUMP OUT 2 TO L));
SII=SEQ; NCR=SI; TALLY=8; GO TRANS; % USE SEQ NO OF THIS CARD
L1
IF SC="" THEN
BEGIN
SII=SI+1; DII=LCR; DSI=1 LIT""; % STOPPER FOR SCAN
NCR=SI; % TEMP, STORAGE, SINCE NCR IS "LOCAL" TO GETVOID,
% (IF SC="" THEN JUMP OUT ELSE
BEGIN TALLY=TALLY+1; SII=SI+1 END);
END
ELSE BEGIN
NCR=SI; % TEMP, STORAGE, SINCE NCR IS "LOCAL" TO GETVOID,
DII=LCR; DSI=1 LIT""; % STOPPER FOR SCAN
% (IF SC="" THEN JUMP OUT ELSE
BEGIN TALLY=TALLY+1; SII=SI+1 END);
END;
TRANS;
SII=NCR; DII=VP; DI=DI+8; % RESTORE POINTERS,
NCR=TALLY; DII=DI-NCR; DSI=NCR CHR;
DI=DI-8; VP=DI; % I.E. "LOC VP"=DI
DI=VR; SII=LOC VP; DSI=WDS; % ADDRESS OF VOID RANGE,
END OF GETVOID;
DOLLARTOG=DOLLARTOG OR TRUE; %
% VOIDT
GETVOID(VOIDPLACE,NCR,VOIDCR,LCR,LASTSEQ); %
% VOIDT
IF TLCR = 0 THEN % FIRST TIME ONLY
TECR=MKABS(TBUFF[9]);
% VOIDT
GETVOID(VOIDPLACE,NCR,VOIDCR,LCR,LASTSEQ); %
% VOIDT
IF Q = "7INCLU" THEN %
BEGIN DOLLARTOG=STARTINCLUDING; GO EXIT END;
IF Q = "7INCLN" THEN %
BEGIN SWITCHIT(NEWINCLBIT); GO AGAIN END; %
TOTALNOI=-(CONV(ACCUM[1],0,COUNT)+1);
% VOIDT 02583001
CHANGESEQ(CARDNUMBER,LCR); % RESTORE SEQ FIELD FOR VOIDRANGE CHECKS
DOLLARTOG=DOLLARTOG AND NOT TRUE; %
CROSSHATCH; %
RTPAREN=CROSSHATCH; %
DOLLARQARD;
% VOIDT 02736001
% VOIDT 02908001
% VOIDT 02999999
BOOLEAN PROCEDURE READINCLUDE;
BEGIN
LABEL EOF; %
LCR=ILCR; MEDIUM="I";
READ(INCLUDE,10,IBUFF[*])(EOF);
IF COMPARE(ILCR,INCLCR)=1 THEN % THIS CARD OUT OF RANGE
EOF; BEGIN
LASTUSED=SAVLASTUSED; MEDIUM="C"; % IN CASE OF CARD ONLY
READINCLUDE=TRUE;

```

```

02359125
02359150
02359175
02359200
02359225
02359250
02359275
02359300
02359325
02359350
02359375
02359400
02359425
02359450
02359475
02359500
02359525
02359550
02359575
02359600
02359625
02359650
02359675
02359700
02359725
02366000
02368000
02410000
02411000
02439000
02456000
02458000
02485000
02486000
02570000
02571000
02572000
02573000
02581000
02582000
02602800
02603000
02640000
02642000
02530000
02732000
02907000
02985001
02986000
02986100
02986200
02986300
02986400
02986500
02986600
02986700
02986800

```

BLANKET(0, LASTSEQ);	02986900
CLOSE(INCLUDE);	02987000
END;	02987100
END READINCLUDE;	02987200
BOOLEAN PROCEDURE STARTINCLUDING;	02987300
BEGIN	02987400
LABEL AWAY, THRU;	02987500
REAL FIRST, T; INTEGER I, J, K;	02987600
BOOLEAN SVNEWTG;	02987700
DEFINE FLAGIT(E) = BEGIN FLAG(E); GO AWAY END#;	02987800
FINDFIRST = SVNEWTG#, MAXR = T#, NEW = FIRST#;	02987900
IF OMITTING THEN GO AWAY;	02988000
IF VOIDING OR VOIDTAPE THEN	02988100
IF COMPARE(MKABS(LASTSEQ), VOIDCR) = 0 THEN *	02988200
IF VOIDTAPE AND LASTUSED = 3 OR NOT VOIDTAPE THEN GO AWAY;	02988300
CHANGESEQ(CARDNUMBER, LCR); * RESTORE FOR LIST	02988400
SVNEWTG = NEWTG;	02988500
NEWTG = NEWTG AND NOT NEWINCL;	02988600
OUTPUTSOURCE; CARDCOUNT = CARDCOUNT + 1; *	02988700
NEWTG = SVNEWTG;	02988800
TURNONSTOPLIGHT("X", LCR); * FOR SCANNING	02988900
STARTINCLUDING = BOOLEAN("H"); * TELL READACARD WE DID OUR THING	02989000
IF LASTUSED = 0 THEN FLAGIT(620); * CANT NEST THEM	02989100
SKAN;	02989200
IF SAVLASTUSED = 0 AND RESULT = 1 THEN FLAGIT(621); * MUST NAME IT FIRST	02989300
IF RESULT = 1 THEN * FILE ID	02989400
BEGIN	02989500
IF COUNT > 7 THEN FLAGIT(622);	02989600
T = " "; MOVECHARACTERS(COUNT, ACCUM[1], 3, T, 1);	02989700
IF INCLUDE, MFID = T THEN INCLUDE, MFID = NEW = T;	02989800
SKAN; IF RESULT = 2 THEN SKAN;	02989900
IF RESULT = 1 OR COUNT > 7 THEN FLAGIT(622);	02990000
T = " "; MOVECHARACTERS(COUNT, ACCUM[1], 3, T, 1);	02990100
IF INCLUDE, FID = T THEN INCLUDE, FID = NEW = T;	02990200
SKAN;	02990300
END;	02990400
INCLRANGE = "99999999"; FINDFIRST = FALSE;	02990500
IF RESULT = 3 THEN * NUMBER IS STARTING NUMBER	02990600
BEGIN	02990700
FINDFIRST = TRUE; FIRST = 0;	02990800
IF COUNT > 8 THEN FLAGIT(623);	02990900
MOVECHARACTERS(COUNT, ACCUM[1], 3, FIRST, 8 = COUNT);	02991000
THRU; SKAN;	02991100
IF Q = "1*0000" OR Q = "2*0000" THEN	02991200
BEGIN * THRU SPECIFIER	02991300
SKAN;	02991400
IF RESULT = 3 OR COUNT > 8 THEN FLAGIT(623); * BAD NUMBER	02991500
INCLRANGE = 0;	02991600
MOVECHARACTERS(COUNT, ACCUM[1], 3, INCLRANGE, 8 = COUNT);	02991700
SKAN;	02991800
END;	02991900
END	02992000
ELSE	02992100
IF Q = "1*0000" THEN * START WHERE WE LEFT OFF	02992200
BEGIN	02992300
IF SAVLASTUSED = 0 OR NEW = 0 THEN FLAGIT(624);	02992400
MOVE(1, Ibuff[9], FIRST); FINDFIRST = TRUE;	02992500

Moore Business Forms, Inc. NY

```

GO TO THRU; % AND GET = SEQ NO
END;
IF Q#"1%0000" OR XMODE#0 THEN FLAGIT(625); % BAD SYNTAX
ILCRI=MKABS(IBUFF[9]);
SEARCH(INCLUDE,IBUFF[*]);
IF MAXR=IBUFF[5]+1 < 0 THEN FLAGIT(626); % FILE NOT ON DISK
IF FINDFIRST THEN % BINARY SEARCH FOR FIRST RECORD
BEGIN
  INCLCRI=MKABS(FIRST);
  KI=MAXR+JI=1;
  WHILE J < K DO JI=J+J; IT=J DIV 2;
  FOR KI=I-1 STEP 1 IF J=0 THEN I ELSE =I
  WHILE I > 0 DO
  BEGIN
    IF K >= MAXR THEN JI=1 ELSE
    BEGIN
      READ(INCLUDE[K],10,IBUFF[*]);
      JI=COMPARE(ILCR,INCLCR);
    END;
    IT=IF J#2 THEN 0 ELSE I DIV 2;
  END;
  IF J#2 THEN BEGIN CLOSE(INCLUDE); FLAGIT(627) END; % NOT THERE
END;
SAVLASTUSED=LASTUSED; LASTUSED:=0; % WE ARE INCLUDING
INCLCRI=MKABS(INCLRANGE);
BLANKET(0, LASTSEQ); %
AWAY;
END STARTINCLUDING;
%
COMMENT#####
FORWARD DECLARATIONS
#####;
MOVE(1, LASTSEQ, LIN[12]);
WRITERROR(REMOTOG, ERRNUM, ACCUM[1], LIN, Q, [12:6], LASTSEQ);
% VOIDT
MOVE(1, LASTSEQ, PNCH(9)); %
IF ERRNUM=199 OR ERRNUM=200 OR %
ERRNUM=611 OR ERRNUM=615 THEN GO TO ENDOFITALL;
LASTUSED=1; % CARD ONLY
TOTALNO:=(1+ADDVALUE:=1000);
% VOIDT
NEXTINFO=LASTINFO+SORTA+1;
BLANKET(1, LASTERR); % BLANK LASTERR AND LASTSEQ
MEDIUM=C; LASTUSED=1; % ASSUME CARD UNTIL TOLD DIFFERE
CLCRI=MKABS(CBUFF[9]); %
READACARD; % READ FIRST CARD AND OPEN FILES
NEXTINFO=LASTINFO+SORTA+1
% VOIDT 09275551
MOVECHARACTERS(4, LASTERR, 0, GT1, 4); %
MOVECHARACTERS(4, LASTERR, 4, GT2, 4); %
% VOIDT 09426001
ERROR(799); % GOES TO ENDOFITALL
TIME:=TIME(1); % BOJ TIME
GETREADY; % READY SET --- GO ---
BLOCK(FALSE); %
ENDOFITALL;
CLEANUP; %

```

```

02992600
02992700
02992800
02992900
02993000
02993100
02993200
02993300
02993400
02993500
02993600
02993700
02993800
02993900
02994000
02994200
02994300
02994400
02994500
02994600
02994700
02994800
02994900
02995000
02995100
02995200
02995300
02995400
03000000
03000100
03000200
03000300
05039500
05044000
05045000
05058000
05107100
05107200
07030000
09028920
09029000
09033000
09034500
09035000
09036000
09037000
09214980
09275100
09415000
09416000
09421000
13312500
17000000
17000100
17800200
17000300
17000400

```

```

S# PATCH NUMBER 422 FOR XALGOL CONTAINS 22 CARDS
S! THIS PROVIDES A BEGIN = END MATCH UP AS ON THE B6700, DHB 11/13/73
  INTEGER NEWBEGCT,OLDBEGCT; % TO MATCH BEGINS = ENDS
S VOID
  EDITLINE(LIN,FCR,ABS(OLDBEGCT),L,[36;10],MEDIUM,OMITTING);
STREAM PROCEDURE EDITLINE(LINE,NCR,BEND,R,SYMBOL,OMIT);
  VALUE
    NCR,BEND,R,SYMBOL,OMIT;
  DSI=LIT" "; NCR=DI; DII=DI-17;
  SII=LOC BEND; DSI=4 DEC; DI:=DI-4; DSI=4 FILL; DII=NCR;
  SII=LOC SYMBOL; SII=SII+7; DSI=CHR; DSI=2 LIT" ";
  OLDBEGCT:=0; % WE PRINTED IT
S VOID 02196611
  IF DEFINCTR = 0 THEN % NOT IN A DEFINE DECLARATION
  IF T.CLASS = BEGINV THEN
    OLDBEGCT:=NEWBEGCT;NEWBEGCT+1 ELSE
  IF T.CLASS = ENDV THEN
  BEGIN
    OLDBEGCT:=IF OLDBEGCT > 0 THEN 0 ELSE =NEWBEGCT;
    NEWBEGCT:=NEWBEGCT-1;
  END;
S VOIDT 0291080;
  EDITLINE(LIN,FCR,0," ",MEDIUM,0);
S VOIDT
S VOIDT
S# PATCH NUMBER 423 FOR XALGOL CONTAINS 15 CARDS
S! THIS CORRECTS AND IMPROVES THE CODE GENERATED FOR CASE STATEMENTS,
S! 1. A CASE INDEX EQUAL TO THE NUMBER OF CASE ELEMENTS WILL NOW
S! CAUSE AN INVALID INDEX AS ALL OTHER INVALID CASE INDICIES DO,
S! 2. FOUR SYLLABLES OF UNNECESSARY BRANCHING CODE IS ELIMINATED AT
S! THE END OF THE CASE STATEMENT,
S! 3. AN EXTRA WORD IN THE CASE BRANCH SEGMENT IS REMOVED,
S! 1/8/74
  ,#415" % LOCAL PATCH LEVEL 1/8/74 DHB
REAL LINK, LLINK, TEMP, N, ADR, PRT, NULL;
LABEL LOOP, XIT, ONOUT;
S VOID 07646421
END ELSE
IF ELCLASS = ENDV THEN % AT ESAC
BEGIN IF N=0 THEN NI=1; % IN CASE OF EMPTY CASE STATEMENT
  IF LINK > 0 AND NOT GOTOG THEN % OMIT LAST BRANCH IF ITS THERE
  BEGIN LINK:=LLINK; LI=L-2 END;
  GO ONOUT;
END;
BEGIN EMIT(LLINK:=LINK); LINK:=LI=L+1 END;
END; %
S VOID 07646551
ONOUT;
S# PATCH NUMBER 424 FOR XALGOL CONTAINS 107 CARDS
S! THIS IMPLEMENTS THE TRUTHSET AS ON THE B6700, IN ADDITION
S! THE DEFAULT TRUTHSET "NUMERIC" IS ADDED,
S! 1/13/74
549 TRUTHSETI; TRUTHSET SPECIFIER MUST BE A STRING,
  TRUTHSETID =118; COMMENT 166;
  EVENTV =23; COMMENT 27;
  TRUTHSETV =24; COMMENT 30;
INTEGER NTSETS; % NUMBER OF TRUTHSETS DECLARED * 3
ARRAY TRUTHSETS[0;119]; % CAN HAVE 40

```

```

00000010
01002400
01002500
02182750
02183500
02183750
02185250
02185300
02185500
02196275
02196510
02869100
02869200
02869300
02869400
02869500
02869600
02869700
02869800
02910100
05038000
14255500
14268500
00000010
00000020
00000030
00000040
00000050
00000060
00000070
01831100
07646390
07646410
07646420
07646500
07646502
07646504
07646506
07646507
07646508
07646509
07646533
07646538
07646540
07646556
00000010
00000020
00000030
00306895
01278075
01298700
01298750
01519000
01520000

```

POWERSOFTEN =(SORTA=1)#;	01574000
SORTA =765#; % MARK XV.2.0 2/13/74	01580000
#424" % LOCAL PATCH LEVEL 1/13/74	01831100
STREAM PROCEDURE DEBUGWORD(SEQ=CODE,FEIL); VALUE SEQ;	04130000
SIT=CODE)	04134000
IF ACLASS=TRUTHSETID AND NOT VBIT THEN GO TO L22; % NO VALUE	07217513
DEFINE CNA = 29#; % CHARACTER IN ALPHA	07957410
CNN = 30#; % CHARACTER IN NUMERIC	07957420
CNT = 31#; % CHARACTER IN TRUTHSETID	07957430
EMIT0(MKS); EMITL(0); EMITL(0); EMITL(0);	07957440
\$ VOID 07957701	07957700
EMITL(0);	07957750
EMITL(CNA); % CHARACTER IN ALPHA	07957800
\$ VOID 07957901	07957900
END ELSE	07958010
IF ELCLASS=TRUTHSETID THEN	07958020
BEGIN	07958030
SIMPLENAME(ELBAT[I]); EMITL(CNT); STEPIT;	07958040
END ELSE	07958050
IF Q="7NUMER" THEN	07958060
BEGIN	07958070
EMITL(0); EMITL(CNN); STEPIT;	07958080
EMITV(CNAT(STRNGCOMPARE));	07958150
SWN = 25#; % SCAN WHILE IN NUMERIC	07961811
SWT = 26#; % SCAN WHILE IN TRUTHSETID	07961812
SUA = 29#; % SCAN UNTIL IN ALPHA	07961820
SUN = 30#; % SCAN UNTIL IN NUMERIC	07961821
SUT = 31#; % SCAN UNTIL IN TRUTHSETID	07961822
BEGIN EMITL(IF WHILECOND THEN SWA ELSE SUA); EMITL(0) END ELSE	07965400
IF ELCLASS = TRUTHSETID THEN	07965410
BEGIN EMITL(IF WHILECOND THEN SWT ELSE SUT);	07965420
SIMPLENAME(ELBAT[I]);	07965430
END ELSE	07965440
IF Q = "7NUMER" THEN	07965450
BEGIN EMITL(IF WHILECOND THEN SWN ELSE SUN); EMITL(0) END ELSE	07965460
\$ VOID 07965601	07965600
TWN = 25#; % TRANSFER WHILE IN NUMERIC	07968410
TWT = 26#; % TRANSFER WHILE IN TRUTHSETID	07968420
TUT = 29#; % TRANSFER UNTIL IN TRUTHSETID	07968430
TUN = 30#; % TRANSFER UNTIL IN NUMERIC	07968440
COMMENT CONDITION IST WHILE IN ALPHA, NUMERIC, TRUTHSETID	07975200
OR; UNTIL IN ALPHA, NUMERIC, TRUTHSETID;	07975300
BEGIN EMITL(IF WHILECOND THEN TWA ELSE TUA); EMITL(0) END ELSE	07975500
IF ELCLASS = TRUTHSETID THEN	07975510
BEGIN EMITL(IF WHILECOND THEN TWT ELSE TUT);	07975520
SIMPLENAME(ELBAT[I]);	07975530
END ELSE	07975540
IF Q = "7NUMER" THEN	07975550
BEGIN EMITL(IF WHILECOND THEN TWN ELSE TUN); EMITL(0) END ELSE	07975560
\$ VOID 07975801	07975800
OCT0132000000560000, "8HAPPE", "NED00000", OCT2000000000100020, %74409214430	
OCT04300000300000000, "8TRUTH", "SET00000",	%758 09214437
OCT00300000200000000, "7PROTE", "CT000000",	%76109214440
OCT20000000000004050, COMMENT POWERS OF TEN ;	%76409214444
0; COMMENT SORTA ;	%76509214445
692 STEP 3 UNTIL 728, 732 STEP 4 UNTIL 748, 750 STEP 2 UNTIL 758,761	09214513
END;	09360100


```

NTSETS=NTSETS+3; 09360200
FOR I=0 STEP 3 UNTIL NTSETS DO 09360300
BEGIN 09360400
    MOVE(1,TRUTHSETS[I],PRT((GT1=TRUTHSETS[I+2]) DIV 128, 09360500
        GT1 MOD 128)); 09360600
    MOVE(1,TRUTHSETS[I+1],PRT((GT1=GT1+1) DIV 128, 09360700
        GT1 MOD 128)); 09360800
    IF BOOARRAYIDSTYPES<INTARRAYID 13330550
        OR TYPE = TASKID 13330560
        OR TYPE = TRUTHSETID 13330570
    THEN % MUST BE CALLED BY NAME 13330580
PROCEDURE TRUTHSET; 13595500
BEGIN 13595510
    STREAM PROCEDURE TRUTHSETTER(N,A,T); VALUE N; 13595520
    BEGIN LOCAL C,CP; 13595530
        S1=A; S1=S1+3; D1=LOC C; D1=D1+7; CP=D1; 13595540
        N ( D1=CP; DS=CHR; D1=T; SKIP DB; SKIP C DB; DS=SET ); 13595550
    END SETTER; 13595560
    LABEL AWAY; 13595570
    IF GTA1[J=1]≠0 THEN FLAG(25); % NO SAVE OR OWN 13595580
    IT=I+1; 13595590
    DO BEGIN 13595600
        STEPIT; P2=TRUE; ENTRY(TRUTHSETID); 13595610
        IF SPECTOG THEN GO AWAY; 13595620
        TRUTHSETS[NTSETS+2]=GETSPACE(TRUE,-12)=1; % CONSTANT OR STRING 13595630
        IF ELCLASS≠LEFTPAREN THEN BEGIN ERR(105); GO AWAY END; 13595640
        IF STEPIT≠STRING AND ELCLASS≠STRNGCON THEN % BAD 13595650
            BEGIN ERR(549); GO AWAY END; 13595660
        DO TRUTHSETTER(COUNT,ACCUM[1],TRUTHSETS[NTSETS]); 13595670
        UNTIL STEPIT≠STRING AND ELCLASS≠STRNGCON; 13595680
        IF ELCLASS≠RTPAREN THEN BEGIN ERR(104); GO AWAY END; 13595690
        IF DEBGTG THEN 13595691
            BEGIN DEBUGWORD(" ",TRUTHSETS[NTSETS],LIN); WRITELINE; 13595692
                DEBUGWORD(" ",TRUTHSETS[NTSETS+1],LIN); WRITELINE; 13595693
            END DEBUG; 13595694
        NTSETS=NTSETS+3; 13595700
    END 13595710
    UNTIL STEPIT≠COMMA; 13595720
AWAY; 13595730
END TRUTHSETDECER; 13595740
    TRUTHSETDEC, % 14016750
        PTRDEC,DEFINDEC,AUXSIZERR,AUXSIZERR,TASKDEC,EVENTDEC, 14021000
        TRUTHSETDEC; 14021250
TRUTHSETDEC: TRUTHSET; GO TO START; 14155800
% PATCH NUMBER 425 FOR XALGOL CONTAINS 4 CARDS 00000000
% THIS ADDS THE NUMBER OF SYNTACTIC ITEMS PROCESSED TO THE COMPILER 00000010
% SUMMARY, I.E, 7700, 7/16/74 DHB 00000020
INTEGER TOKENS; % NUMBER OF SYNTACTIC ITEMS SCANNED, 00504175
    TOKENS=TOKENS+1; % COUNT OF SYNTACTIC ITEMS PROCESSED 02909500
    "PROGRAM CONTAINS",I7," SYNTACTIC ITEMS,"/ 09274455
    TOKENS, % SYNTACTIC ITEM COUNT 09419250
% PATCH NUMBER 501 FOR XALGOL CONTAINS 3 CARDS 14155900
% THIS IS AN IMPROVED VERSION OF PATCH XIV,105, IT COMPILES A FILE 14156100
% DECLARATION SUCH AS FILE FLE 3"ABCDE" ..WITHOUT 14156300
% MAKING THE USE OF OCTAL AND HEX STRINGS IMPOSSABLE IF THE 14156500
% LAST THING ENTERED INTO THE SYMBOL TABLE WAS A FILE, 12/20/72 14156700
% VOID 02761502 % VOID XIV,105 02761500
    IF %OPENRY THEN DEFINECTRI=99; % STOP OCTAL HEX PROBLEM 13080750

```

```

IF DEFINCTR>95 THEN DEFINCTR:=0; % RESET DEFINCTR 13081250
S# PATCH NUMBER 800 FOR XALGOL CONTAINS 35 CARDS
S: THIS PATCH ADDS THE CAPABILITY TO SYNTAX 00000010
S: THE VARIOUS DCP CONSTRUCTS 00000020
S: ERRORBRANCH ATTRIBUTE ADDED 6/5/72 00000030
S: THIS IMPLEMENTS THE COMPILER PORTION OF THE XWRITE ATTRIBUTE, 11/24/72 00000040
28 % 08493094
,"#LASTS" % LASTSTATION 08493121
,"#+0#LASTS" % LASTSTATUS 08493122
,"#7TIME0" % TIMEOUT 08493123
,"#8CHART" % CHARTYPE 08493124
,"#8DCACC" % DCACCESS 08493125
,"#5DCUSE" % DCUSE 08493126
,"#H08USERC" % USERCODE 08493127
,"#+09STNST" % STNSTATUS 08493128
,"#805XREAD" % EXTENDED READ 08493129
,"#B0#ERROR" % ERRORBRANCH 08493130
,"#806XWRIT" % EXTENDED WRITE FOR SYN LINES 08493131
; % END OF FILL STATEMENT (WITH DCP STUFF SAFELY HIDDEN) 08493135
DEFINE NBATTRIBUTES = % THE NUMBER OF STANDARD BIG B ATTRIBUTES 08493501
17# % MARK XV,2,0 2/13/74 08493502
DCPOFFSET = % THE NUMBER OF OVER=LAPPED ATTRIBUTES 08493503
5# % MARK XV,2,0 2/13/74 08493504
DCPATRIBUTENDX= (ATTRIBUTEINDX - DCPOFFSET)#; 08493505
EMITNUM(IF ATTRIBUTEINDX= 1 THEN "6ACCESS" ELSE 08493790
IF ATTRIBUTEINDX= 4 THEN "60THRUS" ELSE 08493791
IF ATTRIBUTEINDX=12 THEN "6ARASIZ" ELSE 08493792
IF DCPATRIBUTENDX=13 THEN "6LSTSTN" ELSE 08493793
IF DCPATRIBUTENDX=14 THEN "6LSTSTS" ELSE 08493794
IF DCPATRIBUTENDX=15 THEN "6TIMOUT" ELSE 08493795
IF DCPATRIBUTENDX=16 THEN "6CHRITYP" ELSE 08493796
IF DCPATRIBUTENDX=17 THEN "6DCACES" ELSE 08493797
IF DCPATRIBUTENDX=20 THEN "6STNSTS" ELSE 08493798
IF DCPATRIBUTENDX=22 THEN "6ERRBRN" ELSE 08493799
IF DCPATRIBUTENDX=23 THEN "6XWRITE" ELSE 08493800
0 & FILEATTRIBUTES[ATTRIBUTEINDX][6112136] 08493820
& FILEATTRIBUTES[ATTRIBUTEINDX][11311]) ; 08493830
EMITL((ATTRIBUTEINDX - % 08493840
(IF ATTRIBUTEINDX > NBATTRIBUTES THEN DCPOFFSET+1 08493842
ELSE 1) 08493844
) & REAL(N#FP OR N#FA)[3914711] 08493846

```

LABEL 00000000LINE 00178104?EXECUTE 0/R

0 /R

Moore Business Forms, Inc. 3V