LABEL 000000000PRINTER00175099CC EX OBJECT/READ;FILE SOURCEFILE=SYMBOL/LOAD;END←

OBJECT /READ

SYMBOL/LOAD

```
BEGIN                                                                   00010000
% LOAD/CANDE SOURCE PROGRAM, REVISED 7/72 (SHM)                         00010100
  COMMENT: * TITLE: B5500/B5700 MARK XIV SYSTEM RELEASE              *  00010110
           * FILE ID: SYMBOL/LOAD  TAPE ID: SYMBOL2/FILE000          *  00010111
           * THIS MATERIAL IS PROPRIETARY TO BURROUGHS CORPORATION    *  00010112
           * AND IS NOT TO BE REPRODUCED, USED, OR DISCLOSED          *  00010113
           * EXCEPT IN ACCORDANCE WITH PROGRAM LICENSE OR UPON        *  00010114
           * WRITTEN AUTHORIZATION OF THE PATENT DIVISION OF          *  00010115
           * BURROUGHS CORPORATION, DETROIT, MICHIGAN 48232           *  00010116
           *                                                          *  00010117
           * COPYRIGHT (C) 1971, 1972 BURROUGHS CORPORATION           *  00010118
           *  AA320206   AA332366   AA386657                          *;  00010119
REAL COMMON;                                                            00010200
SAVE ARRAY A[0:30],BUFFER1,BUFFER2[0:300];                              00010300
SAVE ARRAY N[0:30],DISKROWS[0:19];                                      00010400
REAL APTR, ASTART, EOFPTR, HDRO, FID, I, LINE,                          00010500
LREC, MFID, NCT, NPTR, PREVSEQ, RESULT1, RESULT2, RPB, RSIZE,           00010600
SEQLOC, SPB, SPR, SRCESEQ, USER, WPB;                                   00010700
BOOLEAN BRAAK, CONCISE, OK, CEREAL;                                     00010800
LABEL ERR, EXIT;                                                        00010900
DEFINE EOFMARK=100000000#, MASK=3"2000000000"#;                         00011000
DEFINE MSGSIZE=((APTR.[33:15]-ASTART)*8+APTR.[30:3])#;                  00011100
FILE OUT NEWTAB   DISK SERIAL [20:30 ] (2,30,300,SAVE 1);               00011200
%***********************************************************************00011300
PROCEDURE DISKWAIT(I,A,S,D);                                            00011400
VALUE I,S,D; REAL I,S,D; ARRAY A[*]; COMMUNICATE(-8);                   00011500
%***********************************************************************00011600
PROCEDURE GETHEADER(AREA,NAM1,NAM2,NAM3); VALUE NAM1,NAM2,NAM3;         00011700
REAL NAM1,NAM2,NAM3; ARRAY AREA[*]; COMMUNICATE("21);                   00011800
%***********************************************************************00011900
STREAM PROCEDURE OUTCONV(A,N,SUPRES); VALUE N,SUPRES;                   00012000
%***********************************************************************00012100
   BEGIN                                                                00012200
   SI:=LOC N; DI:=A; DS:=8DEC;                                          00012300
   SUPRES(DI:=DI-8; DS:=7FILL);                                         00012400
   END STREAM PROCEDURE OUTCONV;                                        00012500
%***********************************************************************00012600
STREAM PROCEDURE MOVE(N,A,B); VALUE N;                                  00012700
   BEGIN SI:=A; DI:=B; DS:=N WDS; END;                                  00012800
%***********************************************************************00012900
REAL STREAM PROCEDURE ADDRESS(A);                                       00013000
   BEGIN SI:=A; ADDRESS:=SI; END;                                       00013100
%***********************************************************************00013200
REAL STREAM PROCEDURE INPCONV(A); VALUE A;                              00013300
   BEGIN SI:=A; DI:=LOC INPCONV; DS:=8OCT; END;                         00013400
%***********************************************************************00013500
PROCEDURE TWXOUT(A,N,T); VALUE N,T; REAL A,N,T;                         00013600
%***********************************************************************00013700
   BEGIN                                                                00013800
   COMMUNICATE(-11);                                                    00013900
   BRAAK:=BOOLEAN(T); % MCP RETURNS 1 IF BREAK OCCURRED                 00014000
   APTR:=ASTART;                                                        00014100
   END;                                                                 00014200
%***********************************************************************00014300
STREAM PROCEDURE PLACESTRING(APTR,STRING,CONVERT); VALUE STRING,CONVERT;00014400
%***********************************************************************00014500
   BEGIN LOCAL SV; LABEL EXIT;                                          00014600
   SI:=LOC STRING; DI:=APTR; DI:=DI+5; DI:=DC; SV:=DI;                  00014700
   CONVERT(DS:=8DEC; DI:=DI-8; DS:=7FILL);                              00014800
   SI:=SV; DI:=SV; 8(IF SC=" " THEN SI:=SI+1 ELSE DS:=CHR);             00014900
```

```
    JUMP OUT TO EXIT);                                                          00015000
    SI:=SI+1; DS:=7CHR;                                                         00015100
EXIT:                                                                           00015200
    DS:=LIT" "; SV:=DI; SI:=LOC SV; DI:=APTR; DS:=WDS;                          00015300
    END STREAM PROCEDURE PLACESTRING;                                           00015400
%***************************************************************************00015500
STREAM PROCEDURE MESSAGE(APTR,TYPE); VALUE TYPE;                                00015600
%***************************************************************************00015700
    BEGIN LOCAL SV; LABEL L0,L1,L2,L3,EXIT;                                     00015800
    DI:=APTR; DI:=DI+5; DI:=DC;                                                 00015900
    CI:=CI+TYPE;                                                                00016000
    GO L0; GO L1; GO L2; GO L3;                                                 00016100
L0: % TYPE=0, ERROR FOR NON-STANDARD RECORD SIZE                                00016200
    DS:=37LIT"ERROR: FILE RECORDS ARE NON-STANDARD.";                           00016300
    GO EXIT;                                                                    00016400
L1: % TYPE=1, NUMBER OF RECORDS LOADED                                          00016500
    DS:=34LIT"RECORDS LOADED,LAST RECORD LOADED=";                              00016600
    GO EXIT;                                                                    00016700
L2: % TYPE=2, SEQUENCE ERROR                                                    00016800
    DS:=16LIT"SEQUENCE ERROR: ";                                                00016900
    GO EXIT;                                                                    00017000
L3: % TYPE=3, SYSTEM ERROR                                                      00017100
    DS:=30LIT"SYSTEM ERROR, FILE NOT LOADED.";                                  00017200
EXIT:                                                                           00017300
    SV:=DI; SI:=LOC SV; DI:=APTR; DS:=WDS;                                      00017400
    END STREAM PROCEDURE MESSAGE;                                               00017500
%***************************************************************************00017510
PROCEDURE PARITYERROR(RECORD); VALUE RECORD; REAL RECORD;                       00017515
%***************************************************************************00017520
    BEGIN                                                                       00017525
    % CALLS ON ALGOL-READ INTRINSIC TO OBTAIN PAR-ND LABEL MESSAGE             00017530
    % SO THAT "HELP/DISK" PROGRAM WILL BE CALLED                               00017535
    FILE DISK DISK RANDOM (2,RPB,HDRO.[15:15]);                                 00017540
    FILL DISK WITH MFID,FID;                                                    00017545
        DO BEGIN                                                                00017550
        READ(DISK[RECORD],RPB,A[*]); RECORD:=RECORD+1;                          00017555
        END UNTIL FALSE;                                                        00017560
    END PROCEDURE PARITYERROR;                                                  00017565
%***************************************************************************00017600
PROCEDURE DISKIO(BUFF,SIZE,DISKADDR,RESULT,MASK,WATE);                          00017700
%***************************************************************************00017800
VALUE BUFF,SIZE,DISKADDR,MASK,WATE;                                             00017900
INTEGER SIZE,DISKADDR,BUFF; REAL RESULT,MASK; BOOLEAN WATE;                     00018000
    BEGIN                                                                       00018100
    PROCEDURE IOREQUEST(FINAL,IODESC,LOCATION);                                 00018200
    VALUE FINAL,IODESC,LOCATION; REAL FINAL,IODESC,LOCATION;                    00018300
    COMMUNICATE(41);                                                            00018400
    REAL UNIT,TINU,IOD,DSKADRS;                                                 00018500
    REAL STREAM PROCEDURE ADRS(BUFF,DISKADDR); VALUE BUFF,DISKADDR;             00018600
        BEGIN                                                                   00018700
        SI:=LOC DISKADDR; DI:=BUFF;      DS:=8DEC;                              00018800
        SI:=LOC DISKADDR; DI:=LOC ADRS; DS:=8DEC;                               00018900
        END STREAM PROCEDURE ADRS;                                              00019000
    IF BOOLEAN( (DSKADRS:=ADRS(BUFF,DISKADDR)).[1:5]) THEN                      00019100
        BEGIN UNIT:=19; TINU:=12; END ELSE BEGIN UNIT:=18; TINU:=6; END;        00019200
    IOD:=BUFF&SIZE[8:38:10]&                                                    00019300
    ((SIZE.[38:10]+29) DIV 30+512)[18:33:15]&1[24:47:1]&TINU[3:43:5];           00019400
    RESULT:=ADDRESS(RESULT);                                                    00019500
    IOREQUEST(-IOD&3"347"[25:40:8],IOD,RESULT&UNIT[12:42:6]&1[2:47:1]);         00019600
    IF WATE THEN WAIT(ADDRESS(RESULT),MASK);                                    00019700
```

```
      END PROCEDURE DISKIO;                                                    00019800
%*********************************************************************00019900
BOOLEAN PROCEDURE GETRECORD(RECORD,CEREAL); VALUE RECORD,CEREAL;               00020000
%*********************************************************************00020100
REAL RECORD; BOOLEAN CEREAL;                                                   00020200
   BEGIN                                                                       00020300
   % READS DISK FILE RECORD SEQUENCE FIELD INTO VARIABLE "SRCESEQ"              00020400
   OWN REAL NEXTRECORD,RMIN1,RMIN2,RMAX1,RMAX2,ADRS1,ADRS2;                    00020500
   REAL BUFFER;                                                                00020600
   DEFINE BUFFERSIZE=NEXTRECORD#; % TEMPORARY STORAGE                          00020700
   OWN BOOLEAN SETT,VALID,OK1,OK2;                                             00020800
   BOOLEAN FILL1,FILL2;                                                        00020900
   INTEGER OFFSET;                                                             00021000
   LABEL START,TRANSFER,EXIT;                                                  00021100
   %.....................................................................00021200
   STREAM PROCEDURE MOVE(N,A,B); VALUE N,A;                                    00021300
      BEGIN SI:=A; DI:=B; DS:= N WDS; END;                                     00021400
   %.....................................................................00021500
   PROCEDURE FILLBUFFER(BUFF,RECORD,RESULT,RMIN,RMAX);VALUE BUFF,RECORD;       00021600
   %.....................................................................00021700
   REAL BUFF,RECORD,RESULT,RMIN,RMAX;                                          00021800
      BEGIN LABEL EXIT;                                                        00021900
      REAL BLOCK,ROW;                                                          00022000
      INTEGER SEGMENT,BLOCKADDRESS;                                            00022100
      ROW:=(SEGMENT:=(BLOCK:=RECORD DIV RPB)×SPB) DIV SPR;                     00022200
      IF ROW LEQ 19 THEN                                                       00022300
      IF (BLOCKADDRESS:=DISKROWS[ROW]) NEQ 0 THEN                              00022400
         BEGIN                                                                 00022500
         BLOCKADDRESS:=BLOCKADDRESS+(SEGMENT MOD SPR);                         00022600
         DISKIO(BUFF,(RPB×RSIZE),BLOCKADDRESS,RESULT,MASK,FALSE);              00022700
         RMAX:=(NEXTRECORD:=(RMIN:=BLOCK×RPB)+RPB)-1;                          00022800
         GO TO EXIT;                                                           00022900
         END;                                                                  00023000
      RESULT:=MASK; RMIN:=RMAX:=-1;                                            00023100
EXIT:                                                                          00023200
      END PROCEDURE FILLBUFFER;                                                00023300
   %.....................................................................00023400
   IF NOT SETT THEN                                                            00023500
      BEGIN                                                                    00023600
      RMIN1:=RMAX1:=RMIN2:=RMAX2:= -1;                                         00023700
      SETT:=TRUE;                                                              00023800
      END; % OF INITIALIZE                                                     00023900
   IF RECORD GTR EOFPTR OR RECORD LSS 0 THEN GO TO EXIT;                       00024000
   IF NOT OK1 THEN                                                             00024100
      BEGIN                                                                    00024200
      BUFFER1[0]:=0; ADRS1:=ADDRESS(BUFFER1[0]); OK1:=TRUE;                    00024300
      END;                                                                     00024400
   IF CEREAL AND NOT OK2 THEN                                                  00024500
      BEGIN                                                                    00024600
      BUFFER2[0]:=0; ADRS2:=ADDRESS(BUFFER2[0]); OK2:=TRUE;                    00024700
      END;                                                                     00024800
   IF CEREAL AND NOT VALID THEN FILL2:=TRUE; % READING SERIALLY                00024900
   VALID:=CEREAL;                                                              00025000
START:                                                                         00025100
   IF RECORD GEQ RMIN1 AND RECORD LEQ RMAX1 THEN                               00025200
      BEGIN                                                                    00025300
      BUFFER:=ADRS1;                                                           00025400
      IF NOT BOOLEAN(RESULT1.[19:1]) THEN WAIT(ADDRESS(RESULT1),MASK);         00025500
      IF RESULT1.[28:2] NEQ 0 THEN PARITYERROR(RMIN1);                         00025510
      FILL1:=(RECORD=RMAX1) AND CEREAL;                                        00025600
```

```
            GO TO TRANSFER;                                                    00025700
            END;                                                               00025800
        IF CEREAL THEN IF RECORD GEQ RMIN2 AND RECORD LEQ RMAX2 THEN           00025900
            BEGIN                                                              00026000
            BUFFER:=ADRS2;                                                     00026100
            IF NOT BOOLEAN(RESULT2.[19:1]) THEN WAIT(ADDRESS(RESULT2),MASK);   00026200
            IF RESULT2.[28:2] NEQ 0 THEN PARITYERROR(RMIN2);                   00026210
            FILL2:=(RECORD=RMAX2) AND CEREAL;                                  00026300
            GO TO TRANSFER;                                                    00026400
            END;                                                               00026500
        IF NOT BOOLEAN(RESULT1.[19:1]) THEN WAIT(ADDRESS(RESULT1),MASK);       00026600
        FILLBUFFER(ADRS1,RECORD,RESULT1,RMIN1,RMAX1);                          00026700
        IF CEREAL THEN                                                         00026800
            BEGIN                                                              00026900
            IF NOT BOOLEAN(RESULT2.[19:1]) THEN WAIT(ADDRESS(RESULT2),MASK);   00027000
            IF RESULT2.[28:2] NEQ 0 THEN PARITYERROR(RMIN2);                   00027010
            FILLBUFFER(ADRS2,NEXTRECORD,RESULT2,RMIN2,RMAX2);                  00027100
            END;                                                               00027200
        FILL2:=FALSE;                                                          00027300
        GO TO START;                                                           00027400
    TRANSFER:                                                                  00027500
        OFFSET:= (RECORD MOD RPB)*RSIZE+1;                                     00027600
        SRCESEQ:=INPCONV(BUFFER+OFFSET+SEQLOC);                                00027700
        GETRECORD:=TRUE;                                                       00027800
        IF FILL1 THEN                                                          00027900
            BEGIN                                                              00028000
            IF NOT BOOLEAN(RESULT1.[19:1]) THEN WAIT(ADDRESS(RESULT1),MASK);   00028100
            IF RESULT1.[28:2] NEQ 0 THEN PARITYERROR(RMIN1);                   00028110
            FILLBUFFER(ADRS1,NEXTRECORD,RESULT1,RMIN1,RMAX1);                  00028200
            END;                                                               00028300
        IF FILL2 THEN                                                          00028400
            BEGIN                                                              00028500
            IF NOT BOOLEAN(RESULT2.[19:1]) THEN WAIT(ADDRESS(RESULT2),MASK);   00028600
            IF RESULT2.[28:2] NEQ 0 THEN PARITYERROR(RMIN2);                   00028610
            FILLBUFFER(ADRS2,NEXTRECORD,RESULT2,RMIN2,RMAX2);                  00028700
            END;                                                               00028800
    EXIT:                                                                      00028900
        END PROCEDURE GETRECORD;                                               00029000
    %=============================================================================00029100
    A[0]:=0; APTR:=ASTART:=ADDRESS(A[0]);                                      00029200
    DISKWAIT(1,A,30,COMMON); COMMON:=0; % ESP DISK RECORD                      00029300
    CONCISE         := BOOLEAN(A[1].[8:1]); % CONCISE OPTION SET               00029400
    USER            := A[2]; % USER CODE                                      00029500
    MFID            := A[3]; % FIRST NAME OF INPUT FILE                       00029600
    FID             := A[4]; % SECOND NAME OF INPUT FILE                      00029700
    OUTCONV(LINE,A[1].[40:8],FALSE); % DECIMAL VALUE OF LINE NUMBER           00029800
    TWXOUT(A[0],0,1);                                                          00029900
    A[0]:= -1; GETHEADER(A,MFID,FID,USER);                                     00030000
    IF A[0] LSS 0 THEN                                                         00030100
        BEGIN                                                                  00030200
        I:=3; GO TO ERR; % SYSTEM ERROR                                        00030300
        END;                                                                   00030400
    HDR0        := A[0];                                                       00030500
    RSIZE       := HDR0.[1:14];                                                00030600
    RPB         := HDR0.[30:12];                                               00030700
    SPB         := HDR0.[42:6];                                                00030800
    EOFPTR      := A[7];                                                       00030900
    SPR         := A[8];                                                       00031000
    MOVE(20,A[10],DISKROWS[0]);                                                00031100
    CEREAL:=(EOFPTR GTR 29);                                                   00031200
```

```
      IF (RSIZE NEQ 10 AND EOFPTR GEQ 0) OR (RPB GTR 30) THEN                    00031300
         BEGIN % EMPTY OR NON-STANDARD RECORDS                                   00031400
         I:=0;                                                                   00031500
      ERR: MESSAGE(APTR,I); % NON-STND FILE OR SYSTEM ERROR                      00031600
         TWXOUT(A[0],MSGSIZE,1);                                                 00031700
         COMMON:=0 & 1[2:47:1]; % ERROR FLAG FOR CANDE                           00031800
         GO TO EXIT;                                                             00031900
         END;                                                                    00032000
      SEQLOC:=9;                                                                 00032100
                                                                                 00032200
      RESULT1 := RESULT2 := MASK; % INITIALIZE RESULT DESCRIPTOR                 00032300
      FILL NEWTAB WITH "  "&"1T"[6:36:12]&LINE[18:30:18],USER;                   00032400
         BEGIN % INNER BLOCK                                                     00032500
         LABEL READIN;                                                           00032600
         PREVSEQ := NCT := LREC := -1;                                           00032700
         IF EOFPTR GEQ 0 THEN                                                    00032800
         WHILE GETRECORD((LREC:=LREC+1),CEREAL) DO                               00032900
            BEGIN                                                                00033000
            NCT:=NCT+1; % OUTPUT FILE RECORD COUNT                               00033100
            IF SRCESEQ LEQ PREVSEQ THEN % OUT OF SEQUENCE                        00033200
            IF NOT BRAAK THEN                                                    00033300
               BEGIN                                                             00033400
               MESSAGE(APTR,2); % SEQUENCE ERROR                                 00033500
               PLACESTRING(APTR,PREVSEQ,TRUE);                                   00033600
               PLACESTRING(APTR,SRCESEQ,TRUE);                                   00033700
               TWXOUT(A[0],MSGSIZE,1);                                           00033800
               END;                                                              00033900
            IF NPTR:=NPTR+1 GTR 29 THEN % TAB SEGMENT IS FILLED                  00034000
               BEGIN                                                             00034100
               WRITE(NEWTAB,30,N[*]);                                            00034200
               NPTR:=0;                                                          00034300
               END;                                                              00034400
            N[NPTR]:=0 & NCT[4:32:16] & SRCESEQ[21:21:27];                       00034500
            PREVSEQ:=SRCESEQ;                                                    00034600
            END; % IF MORE RECORDS IN FILE                                       00034700
         IF NPTR:=NPTR+1 GTR 29 THEN % FULL TAB SEGMENT                          00034800
            BEGIN                                                                00034900
            WRITE(NEWTAB,30,N[*]);                                               00035000
            NPTR:=0;                                                             00035100
            END;                                                                 00035200
         N[NPTR]:=EOFMARK;                                                       00035300
         WRITE(NEWTAB,30,N[*]);                                                  00035400
         LOCK(NEWTAB,*);                                                         00035500
         END INNER BLOCK;                                                        00035600
      IF NOT CONCISE THEN                                                        00035700
         BEGIN                                                                   00035800
         PLACESTRING(APTR,NCT+1,TRUE);                                           00035900
         MESSAGE(APTR,1); % NUMBER OF RECORDS LOADED                             00036000
         PLACESTRING(APTR,SRCESEQ,TRUE);                                         00036100
         TWXOUT(A[0],MSGSIZE,1);                                                 00036200
         END; % IF NOT CONCISE                                                   00036300
      COMMON:=COMMON & (NCT+1)[3:3:45]; % PASS RECORD COUNT TO CANDE             00036400
      IF NOT BOOLEAN(RESULT1.[19:1]) THEN WAIT(ADDRESS(RESULT1),MASK);           00036500
      IF NOT BOOLEAN(RESULT2.[19:1]) THEN WAIT(ADDRESS(RESULT2),MASK);           00036600
      EXIT:                                                                      00036700
      END PROGRAM.                                                               00036800
      END;END.          LAST CARD ON ORDING TAPE                                 99999999
```

LABEL 000000000PRINTER00175099CC EX OBJECT/READ;FILE SOURCEFILE=SYMBOL/LOAD;END←

OBJECT /READ