

LABEL 000000000PRINTER00175099CC EX OBJECT/READ;FILE SOURCEFILE=SYMBOL/TAPEDSK;END+

OBJECT /READ

SYMBOL/TAPEDSK

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57

Data Documents/Inc.

```

      B=5500 TAPE TO DISK ROUTINE   MAY 1969 BY P.E.G.           00020000
COMMENT: * TITLE: B5500/B5700 MARK XIV SYSTEM RELEASE          * 00020100
          * FILE ID: SYMBOL/TAPEDSK TAPE ID: SYMBOL2/FILE000    * 00020101
          * THIS MATERIAL IS PROPRIETARY TO BURROUGHS CORPORATION * 00020102
          * AND IS NOT TO BE REPRODUCED, USED, OR DISCLOSED    * 00020103
          * EXCEPT IN ACCORDANCE WITH PROGRAM LICENSE OR UPON  * 00020104
          * WRITTEN AUTHORIZATION OF THE PATENT DIVISION OF     * 00020105
          * BURROUGHS CORPORATION, DETROIT, MICHIGAN 48232     * 00020106
          *                                                       * 00020107
          * COPYRIGHT 1971, 1972 BURROUGHS CORPORATION          * 00020108
          * AA320206 AA386657                                   *; 00020109

```

```

BEGIN
  DEFINE M=MEMORY#,P=POLISH#;
  DEFINE FILEBASE=38000#;%DEFAULT MCP ADDRESS USED IF 0 IN COLD START
  LABEL TIMER,INN,START,FOUND,GOTCHA,FIRST;
  BOOLEAN B;
  INTEGER J;
  INIEGER K;
  REAL TAPE,INDX,CARD,LOADED,DISK,CHECK,I;
  INTEGER ADDR;
  ARRAY A[*],BUFFER[*];
  ARRAY LOADBUTTON[30];

```

```

      OCT0441000401570421, % 1 20 00090000
      OCT0157000000104411, % 2 21 00100000
      OCT0211001441310055, % 3 22 00110000
      OCT4155124500004425, % 4 23 00120000
      OCT0060013101600064, % 5 24 00130000
      OCT4441010402530305, % 6 25 00140000
      OCT0100023441310055, % 7 26 00150000
      OCT0062011441310055, % 8 27 00160000
      OCT0066013441310055, % 9 30 00170000
      OCT0072015441310055, % 10 31 00180000
      OCT0076017441310055, % 11 32 00190000
      OCT5140000047700200, % 12 33 00200000
      OCT0441100401004441, % 13 34 00210000
      OCT0253010420527405, % 14 35 00220000
      OCT7405005101002411, % 15 36 00230000
      0,0,0,0,0; % 16 37 = 43 00240000

```

```

ARRAY MESSAGE[28]:=
  "DISK FAI", "LURE<000", %0 00260000
  "TAPE FAI", "LURE<000", %1 00270000
  "NO MCP F", "ILE<0000", %2 00280000
  "MCP FILE", " LOADED<", %3 00290000
  "NOT IN D", "IRCTORY<", %4 00300000
  "LOADER O", "N DISK< ", %5 00310000
  "CRA FAIL", "URE< ", %6 00320000
  "SPO FAIL", "URE< ", 00321000
  "DISK ROW", " LENGTH=", 00322000
  0, "NEW MCP", 00323000
  "=", " . ENTER", 00324000
  " OK TO C", "ONTINUE<", 00325000
  0,0; 00330000

```

```

ARRAY TABLE [5]:= "MCP " ,"DISK " ,"SYSTEM " ,0,0; 00340000
SAVE REAL PROCEDURE EOM; FORWARD; 00350000
SAVE PROCEDURE KEY(I); VALUE I; REAL I; FORWARD; 00360000
SAVE REAL PROCEDURE IOERR(IOD,R); 00370000
  VALUE IOD; 00380000
  REAL IOD,R; 00390000
  FORWARD; 00400000
SAVE REAL PROCEDURE IO(ADDRESS,IOD,MASK); 00410000

```

Data Documents/Inc.

	VALUE ADDRESS,IOD,MASK;	00420000
	REAL ADDRESS,IUD,MASK;	00430000
	BEGIN LABEL L;	00440000
1	REAL R,T;	00450000
2	IOD ← ADDRESS OR IUD;	00460000
3	L: P([IOD],IIO);	00470000
4	R ← EOM;	00480000
5	T ← R AND NOT MASK;	00490000
6	IF T.[26:7] ≠ 0 THEN	00500000
7	IF NOT T.[29:1] THEN	00510000
8	IF IOERR(IOD,R) THEN GO TO L;	00520000
9	IO ← R;	00530000
10	END;	00540000
11	SAVE PROCEDURE DISKWRITE(LENGTH);	00550000
12	VALUE LENGTH;REAL LENGTH;	00551000
13	BEGIN LABEL L;	00560000
14	REAL J,K;	00570000
15	J:=@156040003600000&LENGTH[27:42:6];	00580000
16	L: P(IO(DISK,J,0),DEL);	00590000
17	M(CHECK) ← M(DISK);	00600000
18	P(IO(CHECK,J&1[24:47:1],0),DEL);	00610000
19	STREAM(DISK:X:=LENGTH,CHECK);	00620000
20	BEGIN SI ← DISK; SI ← SI+8; DI ← DI+8;	00630000
21	X(5(IF 48 SC NEQ DC THEN TALLY := 1));	00640000
22	DISK ← TALLY; END;	00650000
23	IF P THEN	00660000
24	BEGIN K ← K+1;	00670000
25	IF K ≤ 5 THEN GO TO L;	00680000
26	KEY(0);	00690000
27	END;	00700000
28	STREAM(DELTA←"30",DISK);	00710000
29	BEGIN SI ← LUC DELTA;	00720000
30	DS ← 8 ADD;	00730000
31	END;	00740000
32	END;	00750000
33	SAVE PROCEDURE SPACETAPE(N); VALUE N; REAL N;	00760000
34	BEGIN REAL I;	00770000
35	FOR I ← 1 STEP 1 UNTIL N DO	00780000
36	DC UNTIL IO(TAPE,@140000000,@6000000).[27:1];	00790000
37	END;	00800000
38	SAVE REAL PROCEDURE IOERR(IOD,R);	00810000
39	VALUE IOD;	00820000
40	REAL R,IOD;	00830000
41	BEGIN REAL K;	00840000
42	LABEL EXIT;	00850000
43	IF R.[30:3] ≠ 0 THEN P(1,RTN);	00860000
44	IF R.[28:1] THEN	00870000
45	IF R.[3:5] = 30 THEN	00871000
46	BEGIN	00872000
47	FOR K:=1 STEP 1 UNTIL 5 DO	00873000
48	IF(R:=IO(0,IOD,@2000000)).[28:1]=0 THEN	00874000
49	GO EXIT;	00874500
50	END ELSE	00875000
51	BEGIN IF R.[3:5] THEN	00880000
52	BEGIN	00890000
53	FOR K ← 1 STEP 1 UNTIL 5 DO	00900000
54	BEGIN P(IO(TAPE,@340000000,	00910000
55	@6000000),DEL);	00920000
56	IF(R+IO(0,IOD,@6000000)),	00930000
57	[28:1] = 0 THEN	00940000

		GO TO EXIT;	00950000
	END;	END	00960000
	ELSE BEGIN		00970000
1		FOR K + 1 STEP 1 UNTIL 5 DO	00980000
2		IF (R+IO(0,IOD,@2000000))	00990000
3		.[28:1] = 0 THEN GO EXIT;	01000000
4		END END;	01010000
5		KEY(IF R.[3:5]=10 THEN 6 ELSE	01020000
6		IF R.[3:5]=30 THEN 7 ELSE IOO.[7:1]);	01021000
7	EXIT: END;		01030000
8	SAVE PROCEDURE KEY(I); VALUE I; REAL I;		01040000
9	BEGIN P(IO([MESSAGE[2*I]] INX 0,@7400000000000000,0),DEL);		01050000
10	DO UNTIL I = 3;		01060000
11	END;		01070000
12	SAVE INTEGER PROCEDURE PACKER;		01080000
13	BEGIN;		01090000
14	STREAM (A:=[BUFFER[9]],B:=[PACKER]);		01100000
15	BEGIN SI := A; SI := SI + 4;		01110000
16	DI := B; DI := DI + 6;		01120000
17	4(SKIP 3 SB; 3(IF SB THEN DS:=SET ELSE SKIP DB;SKIP SB));		01130000
18	END;		01140000
19	END;		01150000
20	SAVE PROCEDURE LOADER;		01160000
21	BEGIN LABEL TOPPER;		01170000
22		REAL TEMMP;	01180000
23		IF LOADED THEN KEY(5);	01190000
24	TOPPER: STREAM(B:=[BUFFER[1]],A:=BUFFER[9].[6:9],		01200000
25	D+PACKER INX BUFFER);		01210000
26	BEGIN SI +B; DS + A WDS; END;		01220000
27	IF P(RRR).[24:1] THEN %IF CRA READY THEN		01230000
28	P(IO(BUFFER INX 0,@240120040000000, @3000000),DEL)		01240000
29	ELSE BUFFER[0] := @14;		01250000
30	IF BUFFER[0].[4:44]=0 THEN GO TO TOPPER;		01260000
31	TEMP := DISK;		01270000
32	DISK := BUFFER INX 111;		01280000
33	LOADED := TRUE;		01290000
34	M(DISK) := 1;		01300000
35	DISKWRITE(30);		01310000
36	DISK := TEMMP;		01320000
37	END;		01330000
38	SAVE PROCEDURE SCANNER;		01340000
39	BEGIN;		01350000
40	STREAM (BO:=0:I,D:=[TABLE[4]]);		01360000
41	BEGIN		01370000
42	SI := 1;		01380000
43	DI := DI + 1;		01390000
44	DEBLANK: 63 (IF SC = " " THEN SI := SI + 1 ELSE JUMP OUT);		01400000
45	IF SC = "/" THEN		01410000
46	BEGIN		01420000
47	ABLE: SI := SI + 1;		01430000
48	GO TO DEBLANK;		01440000
49	END ELSE		01450000
50	IF SC = "=" THEN GO TO ABLE ELSE		01460000
51	7 (IF SC = " " THEN DS := LIT " " ELSE		01470000
52	IF SC = "/" THEN DS := LIT " " ELSE		01480000
53	IF SC = "=" THEN DS := LIT " " ELSE DS:=CHR);		01490000
54	BC := SI;		01500000
55	END;		01510000
56	I := P;		01520000
57	END SCANNER;		01530000

SAVE PROCEDURE CORRECT (TAPPE);	01540000
VALUE TAPPE; BUCLEAN TAPPE;	01550000
BEGIN	01560000
1 SCANNER;	01570000
2 IF TAPPE THEN TABLE[2] := TABLE[4]	01580000
3 ELSE BEGIN	01590000
4 TABLE[0] := TABLE [4];	01600000
5 SCANNER;	01610000
6 TABLE [1] := TABLE [4];	01620000
7 END;	01630000
8 END CORRECT;	01640000
9 SAVE PROCEDURE DIRECTORYSCAN;	01650000
10 BEGIN LABEL OWT;	01660000
11 INTEGER DIRECTOP;	01670000
12 BUFFER[0] := 0;	01680000
13 P(IO(BUFFER INX 0,@140000040100000,0),DEL);	01690000
14 DIRECTOP := BUFFER [2];	01700000
15 FOR ADDR ← DIRECTOP + 4 STEP 16 WHILE TRUE DO	01720000
16 BEGIN; STREAM (ADRS:=[ADDR],BUFFER);	01730000
17 BEGIN SI := ADRS;	01740000
18 DS := 8 DEC;	01750000
19 END;	01760000
20 P(IO(BUFFER INX 0,@140000042000000,0),DEL);	01770000
21 FOR INDX := 0 STEP 2 UNTIL 28 DO	01780000
22 BEGIN IF (BUFFER[479-INDX] EQV @114) =NOT 0 THEN KEY(4);	01790000
23 IF (BUFFER[479-INDX] EQV TABLE[0])=NOT 0 THEN	01800000
24 IF (BUFFER[480-INDX] EQV TABLE[1])=NOT 0 THEN GO OWT;	01810000
25 END; END;	01820000
26 OWT: IF B+((TABLE[3]+BUFFER[431-INDX*15])=0) THEN TABLE[3]+FILEBASE;	01830000
27 STREAM(A:=TABLE[3], B:=[MESSAGE[24]]);	01840000
28 BEGIN SI := LOC A; DS:=8 DEC; END;	01850000
29 K:=BUFFER[429-INDX*15];	01855000
30 END;	01860000
31 SAVE PROCEDURE FINDMCP;	01870000
32 BEGIN	01880000
33 IF P(RRR).[24:1] THEN	01890000
34 BEGIN	01900000
35 DO	01910000
36 BEGIN	01920000
37 I := P(BUFFER INX 0);	01930000
38 P(IO(BUFFER INX 0,@24012004000000,@7400000),DEL);	01940000
39 IF (*[BUFFER[0]]EQV 0)=NOT 0 THEN LOADER;	01950000
40 SCANNER;	01960000
41 IF (TABLE[4] EQV "FILE ")=NOT 0 THEN	01970000
42 CORRECT(FALSE);	01980000
43 IF (TABLE[4] EQV "TAPE ")=NOT 0 THEN	01990000
44 CORRECT(TRUE)	02000000
45 END UNTIL (P(RRR).[24:1]=0	02010000
46 OR ((TABLE[4]EQV"STOP ")=NOT 0));	02011000
47 END;	02020000
48 DIRECTORYSCAN;	02030000
49 END;	02040000
50 SAVE PROCEDURE UPDATEDIRECT;	02050000
51 BEGIN	02060000
52 A[11] := TABLE [3];	02070000
53 A[ 5] := P(DUP,LOD,SSP); %CLEAR SIGN BIT SO MCP WONT REMOVE	02080000
54 STREAM(AB:= ADDR +((28-INDX)DIV 2),A);	02090000
55 BEGIN SI:=LOC AB;DS:=8 DEC; END;	02100000
56 P(IO(A INX 0,@140360100100000,0),DEL);	02110000
57 BUFFER[0] := @ 40; %FIND SYSTEM NUMBER	02120000



```

FOR I ← 1 STEP 2 UNTIL 31 DO
BEGIN
TAPE := (A INX 1)& 1[3:43:5];
IF IO(TAPE, @17770540000000, @6500000), [26:7] = 0 THEN
IF ((*[A[1]]) EQV (-FLAG("LABEL ") = NOT 0) AND
((*[A[2]]) EQV TABLE[2]) = NOT 0) AND
((*[A[3]]) EQV "FILE000") = NOT 0) THEN GO TO FOUND;
P(IO(TAPE, @4200000000, @500000), DEL);
END
FOUND : UNTIL FALSE;
SPACETAPE (1);
P( IO(TAPE, @17770540000000, 0), DEL);
FOR I ← 1 STEP 2 UNTIL 512 DO
IF ((A[I] EQV TABLE[0]) = NOT 0) AND
((A[I+1] EQV TABLE[1]) = NOT 0)
THEN GO TO GOTCHA
ELSE IF A[I] = @14 THEN KEY(2);
GOTCHA:
SPACETAPE (I+1);
P( IO(TAPE, @17770540000000, 0), DEL);
J:=A[9];
IF J GTR K THEN ASK;
UPDATEDIRECT;
ADDR:=A[9] DIV 30;
FOR I ← 1 STEP 1 UNTIL ADDR DO
BEGIN
P( IO (TAPE, @17770540000000, 0), DEL);
DISKWRITE(30);
END;
IF (ADDR:= J MOD 30) > 0 THEN
BEGIN
P( IO (TAPE, @17770540000000, 0), DEL);
DISKWRITE(ADDR);
END;
P( IO (TAPE, @4200000000, 0), DEL);
IF B THEN BEGIN;
STREAM(A+[TABLE[3]], X+[MESSAGE[6]]);
BEGIN DI+DI+15; DS+20 LIT " TO DEFAULT ADDRESS=";
SI+A; DS+8 DEC; DS+LIT "+";
DI+DI-9; DS+8 FILL;
END;
END;
IF J GTR K THEN
BEGIN; STREAM(A:=(J-K), B, X:=[MESSAGE[6]]);
BEGIN
DI:=LOC B; DS:=8 LIT " TO DEFAU";
SI:=LOC B; DI:=X; DI:=DI+15;
IF 8 SC=DC THEN DI:=DI+20 ELSE DI:=DI-8;
DS:=2 LIT " (";
SI:=LOC A; DS:=8 DEC;
DS:=13 LIT " SEG QVRFLW)+";
DI:=DI-21; DS:=8 FILL;
END;
END;
KEY (3);
STREAM(LOADBUTTON, A+@20); BEGIN SI+LOADBUTTON; DS+20 WDS END;
P(O, STS, O, STF);
GO TO FIRST;
DO UNTIL FALSE;

```

```

02500000
02510000
02520000
02530000
02540000
02550000
02560000
02570000
02580000
02590000
02600000
02610000
02620000
02630000
02640000
02650000
02660000
02670000
02680000
02690000
02700000
02701000
02702000
02710000
02720000
02730000
02740000
02750000
02760000
02770000
02771000
02772000
02773000
02774000
02775000
02780000
02781000
02781100
02781200
02781300
02781400
02781500
02781600
02782000
02782100
02782200
02782300
02782400
02782500
02782600
02782700
02782800
02782900
02783000
02783100
02790000
02800000
02810000
02820000
02830000

```



LABEL 000000000PRINTER00175099CC EX OBJECT/READ;FILE SOURCEFILE=SYMBOL/TAPEDSK;END+

OBJECT /READ

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57