FLEX USER GROUP
NEWSLETTER

3540 STURBRIDGE COURT
ANN ARBOR, MI 48105
ISSUE 2

CORRECTIONS

First things first. Cal Rasmussen phoned all the way from Idaho to tell me about my stupidity (my description, not his). It seems that since I have an ADM-3, I really didn't look at his program carefully. Any of you who did, probably realize that it does not provide a Home-up and Screen Clear as I indicated, but as is clearly indicated in the listing, it provides an erase to the end of line whenever a carriage return linefeed is provided by the FLEX subroutine. The CT-1024 operates in a "page" mode and when it fills a page, it "flips" to the other, displaying old information. Without the Erase-EOL a short line is followed by old "garbage". The Erase-EOL cleans up everything after the end of each new line as it is written.

Sorry Cal, next time, I'll read more carefully. Also, near the end of my notes of this program, I referenced John, rather than Cal.

I should point out here that SWTP published a fix for the CT-1024 that involves adding one wire. It causes the new page to erase when the display pages. There is one problem with their solution. It doesn't work reliably at 1200 baud, usually leaving 3 or 4 lines of one character at the bottom of the page. It works fine, however, at 300 baud. Before I got into trying to solve that one, I added the ADM-3 to my system. Anyone who wants the details from SWTPC let me know, and I'll send a copy to you. Has anyone modified a CT-1024 for scrolling? If so, maybe the others would appreciate a description of "how to".

STRUBAL

I finally managed to get my copy of STRUBAL, a "STRUctured BAsic Language". It took two months from the day that I mailed the cashier's check with the order, but that's another story.

STRUBAL is a compiler, somewhat like A/BASIC. It has the added features of full floating point arithmetic and a full set of scientific functions. It also has nice features that allow for better structured programming:

1. Use of labels rather than line numbers.
2. Variable names up to 6 characters.
3. A DO-WHILE structure for loop control.
4. A FOR-NEXT structure like BASIC.
5. Provisions for full comment lines.
6. The compiler passes source listing formatting such as indenting of a loop, along to the listing at compile time.
7. A rather elegantly simple overall structure that makes the

listing look pleasing and easy to follow.

What followed here in the original manuscript of this Newsletter was a description of my application of STRUBAL to a program that I had written in BASIC and assembler. I had several unkind things to say about some of the features of STRUBAL and the problems I had with it. In the course of my trying to make use of it, I wrote a letter to Hemenway Associates complaining about several "features" of STRUBAL. The evening after I had written this, I received a call from Jack Hemenway in which he explained that I had gotten an early version of STRUBAL, which he is not supporting for FLEX. He has a new version called STRUBAL PLUS. STRUBAL PLUS has most all of the problems resolved that I had inquired about in my note to Hemenway Associates. Indeed, Mr. Hemenway agreed with most if not all of my critical comments. At this point he indicated that he is sending me a copy of STRUBAL PLUS to try out. He indicated that he would like a critical review of it by a person who is at least initially hostile. How could I ask for more?

Hopefully, by next month, I will have results and will be able to give you a hint as to the usefulness of STRUBAL PLUS. The initial reactions are very mixed. The largest problem I see with it is that it is very inefficient. I'm afraid that will be a problem with any compiler for the Microprocessor based computers.

LATE NOTE ON STRUBAL

I've received STRUBAL + and just started to evaluate it. There are some major improvements in logistics. The new Linkage Editor is a step forward in results, but a step backwards in convenience.

I found a few bugs introduced in the fundamental operations, probably in the process of making the additions. I think this is going to be excellent software. Hold off for a while though, it is not yet finished software! Mr. Hemenway must have been impressed with my comments; he has asked me if i would mind reviewing all his new software before it is released. Needless to say, I'll pass my final review and comments along to this group.

RANDOM NOTES

John Jordan wrote again with some comments and some utilities. First of all, the comments. John has a the DMAF large disk system. John, consider this to be at least a partial answer to your questions and offer. John has offered to provide me the service of translating the newsletter to large disk format via a Kansas City format tape that I could provide him. John has also offered to translate the mini-flex utilities for the large system. To my knowledge, John, only one other of the group has the DMAF1, Ron Mahon in Masontown, PA. The name and address list of all those now receiving this letter is included here. Perhaps you two can get together. John I can provide the list and description of the mini-flex routines that you would need to do the translation. I assume that you have the Advanced Programmer Guide for

the large FLEX now.

At this point four people have responded to my question about a disk issue by sending me a disk to be used for this purpose. Please let me know of your interest. I'll need more than four replies if we are to do this. Also, at this point, I've received a total of seven subscription payments. I'm sending this to all again, in the hopes that we can keep things going until the group grows a bit more. If you are still interested, please do so indicate by sending your check. Since this is an informal group, there is no bank account yet in the name FLEX USER'S GROUP. So please make it easy on me by making your check payable to RONALD W. ANDERSON.

UTILITIES BY JOHN JORDAN

The programs identified as numbers 1,2,and 3 are from John. Since john's listings were for a large flex system, I decided to enter them in my system, substuting the mini-flex routine addresses for the equates, and the FLEX utility area starting at $7600 rather than the large FLEX area at $al00. I found only one problem, the fact that mini-FLEX does not have an OUTADR subroutine. I simply wrote one to add to the end of John's programs. It is 4 lines long, and uses OUTHEX The notes that I have added to the listings should allow anyone to use these for either FLEX. Program 2 is a useful memory map program. If you use the clear memory program and then load some files from disk, the MAP command will tell you the start and end address of all the blocks of memory that have been loaded as explained on John's listing, Program 3 is a Dump program that allows Hex and ASCII dump of memory. This is nicely formatted, and allows several options to be specified. Note that the first two of these utilities may be located in the mini-FLEX utility area at $7600 by using this as the ORG. The dump program is too large to fit this area in mini-FLEX so you'll have to put it near the top of your memory, just below FLEX if you have 32K like my system has. John, these are really nice. Thanks for sending them for all of us to share.

UTILITIES FROM JIM MCVAY

Jim sent us three utilities too, two of which are included here as programs 4 and 5. Jim apparently thought the idea I had for a title file to be a good one. He implemented its use very nicely. Program 4, called RETITLE is just that. It allows you to delete a title file and then enters the BUILD utility so that you can enter a new TITLE file. The listing contains some pertinent comments.

Program 5 is one that complements the RETITLE program. It is yet another CATALOG program that Jim calls CAT2.CMD. It reads and lists the TITLE file from the specified disk and then outputs a catalog listing containing just the filenames and extensions in three wide format. Jim indicates that he has added plastic envelopes to the front of each of his disk jackets, and that these catalog listings fit them nicely. There is a minor complication in using FLEX as a subroutine

and maintaining the PRINT mode. If you simply use P,CAT2 you will find that the TITLE file is printed, and the catalog is output to the terminal. You may recall that in a multiple command line, P, is only honored for the command where it is prefixed. Jim has solved the problem internally in his program, but it requires special syntax for the command. Jim didn't provide for the comma separator, only the space. If this bothers you, you may easily modify the routine that now only checks for space.

Jim saves the print status in his program, but you must add the P after the cat2. You must also be certain that you have used the Print routine or at least loaded it by using the P, prefix since having powered up your system. To be safe, you should use P,CAT2 P to print the catalog with the title for drive 1 (assuming that is your working drive.) You may also specify the drive as in P,CAT2 Pl or P0. When I tried this utility, which Jim was kind enough to send me on a disk, it didn't work. I called Jim to ask if he had anything peculiar about his system. We double checked, and it ran for him and not for me. A half hour after talking to him, I was looking at it and realized that he had ORGed it at $100. At this point I realize why it didn't work on my system. It calls the LIST utility, which loads at $100, right over the CAT utility in MY SYSTEM ONLY. I had recently modified the FLEX LIST utility to provide optional paging and a page header. It grew too large for $7600 so I moved it to $100! When I re ORGed the utility at $800 it worked immediately. You may want to ORG it at $l00 as Jim did. Note that you will have to supply the address of the entry of your print routine (the one you use for your PRINT.SYS file. I have spent a couple of evenings entering and verifying these utilities. If anyone out there wants to modify Jim's utility to make the command format absolutely standard, feel free to do so, and we'll publish the improved version next time. Jim indicated to me that he has the bug very badly, and is going to night school in computer science, bud doesn't consider himself an expert programmer at this point.

I'd like to thank both of you, John and Jim for taking the time to pass these along to the rest of us.

DOS AS A SUBROUTINE IN BASIC

Richard Cagle has sent this BASIC program as an example of how to use DOS commands in a BASIC program as subroutines. He describes the program well, so I'll quote his letter. "...It is a simple program that can be used in a BASIC program to call up FLEX as a subroutine. Flex will carry out the command and will return to BASIC provided the command wasn't loaded into the same region that BASIC occupies. (ie. it will not return to BASIC for APPEND, CAT, COPY, NEWDISK, BACKUP,[OR LIST].

Lines 10,20, and 90 should be deleted if using it as a subroutine. The subroutine call is "gosub 40" and of course you need a return at the end.

Line 30 is a subroutine to POKE all of the necessary items into ram. Lines 55 thru 70 simply step through the command line one character at a time - loading it into the DOS buffer. Line 85 transfers control to DOS at a location that returns. Line 40 puts carriage return into the line buffer after the command line, and the last part of line 75 and the first part of line 85 set up the USER(X) pass over."

Richard also made some comments in his letter, one of which was a vote against a "disk" issue of the newsletter. His reasons include differences in our systems requiring changes in programs etc, plus the logistics of getting disks back and forth. I do agree on the logistics problem, but as far as the differences are concerned, you will each be "printing your own" using your own system and print routine. As far as the differences are concerned, isn't it easier to edit a source file on a disk than to have to input the whole thing from a typed copy?? I realize that here, I am asking for more work and the opportunity to give my disk drives a great deal of extra. wear and tear. I had in mind for each of you to specify the width of your printer or terminal, and if necessary to write you a file of the proper format for your printer. I have not yet written the software necessary for the output to a file rather than the printer, but it would be similar to the PRINT.SYS file invoked by the P, command. It would be essentially a WRITE.SYS file invoked by a W, command with the added feature of allowing the specification of a file name. I really don't see that to be too large a problem. Richard indicated that he would go along with the majority on that matter.

Richard also asked that we take a survey of our members as to their interests. He is using his system in a business application, and has developed some programs for himself. He indicates that he has spent a great deal of time doing these, and doesn't feel that they should be given away. I most certainly agree with that. Major software writing involves a major investment in time and effort, and should be rewarded. Randy Lewis and I are putting together a package of utilities now, tha t we plan to advertise and sell. If you will re-read my comment last time, you will find that I said only that I would like to keep such activities separate from the newsletter so that people don't get the idea that this was set up simply as a means to sell software. If you have software for sale, I will pass that information along to our readers with a description of what you have and information on how to contact you for further information and prices. Consider this to be notification of Richard's business software. His address will be found in the list that is part of this issue.

NEWS FROM TSC

I talked with Dan Vanada from TSC today. He is going to give the group a plug in their first TSC FLEX NEWSLETTER. Maybe we can get our group size up to the point where we can get our letter printed and save some of the cost. Dan also indicated that they have FLEX 2

operating. In case you haven't heard, FLEX 2 is a new system for miniFLEX, that includes all the features of the large disk system. They are now finishing up the documentation, and will be shipping FLEX 2 in about 2 weeks. The price is $75. Hopefully by next issue we can review it. TSC also is about to release its new BASIC. This BASIC started out to be a compiler, but they found the "runtime" package growing unreasonably to be almost as large as the interpreter, so they compromised. The end product is not an interpreter and not a compiler, but something in between. We will be reviewing it completely as soon as it is available. Meanwhile I'd better not say too much and give you misinformation.

A number of you have asked me about the BASIC Renumber program that I mentioned in my letter to Kilobaud to which you all responded. Between the time I wrote the letter and its publication, I sent the RENUM program to TSC and they purchased rights to it. Dan Vanada told me yesterday that they will publish it in issue 1 of their FLEX newsletter. When that has happened, it will be in the "public domain". At that time, if you are a paid up FLEX User's Group member, and you send me a blank disk with return postage and some proof that you received the TSC Newsletter I will put a copy of the source listing and binary file on your disk for you. I will also give you source and binary files of all the programs we have published here to date.

CROSS ASSEMBLER

If any of you can get a copy of EDN magazine for Feb. 5, you will find an interesting article by Jack Hemenway in which he indicates how to write a cross assembler for the 8086 processor, that runs on the 6800. He uses his Macro assembler to write the macros that make the assembler operate. It is an interesting technique and one that could be used to write a cross assembler for nearly any processor.

LAST BUT NOT LEAST

Yesterday I received a disk from Garry Caudell containing three utulities. (That seems to be a magic number, so far most all of those submitting utilities have sent three.) After my little self induced difficulty with Jim McVay's utility, I decided that I would in general verify all programs sent for publication. If this issue is to get out by the end of February, I will not have time to include all of Garry's offerings. He sent a very nice SEARCH program that is included here. It allows you to search a specified block of memory for a string of ASCII or Hex characters of any length, and reports the addresses of all occurrences of that string. The program is entirely self prompting. It asks for the start and stop addresses, and then gets the type of string from you. A response of H for Hex or A for ASCII is all that is required. The string is entered, followed by return, and the search starts immediately.

I had written a simple 1 to 3 Hex byte search program several weeks ago while trying to adapt a program to FLEX, but this one is much nicer in capabailty and ease of use. Thanks Garry, I'll try to include your other offerings next month.

FINAL COMMENTS

You will note that this issue is substantially larger than the first. It will continue to grow as you send material for it. You might consider making my job easier for me by sending your programs on a disk. Please use at least a piece of corrugated cardboard on each side for a stiffener. A couple of disks have arrived in marginally usable condition. I promise to return all disks sent to me, either as a User Notes issue, or as an extra. I might let them accumulate for a while until we get a vote from most of you. Enjoy these utilities and reviews. See you next month. Please, if you intend to send me some material, make it as soon as possible so we can keep this as close to monthly as possible.

LATE NOTE

Printing has been held up a couple of days due to a copier being out of dispersant. Please accept my apologies for the lateness of this. I am beginning to feel a bit overwhelmed with the responses from all of you. It will not be possible for very long for me to acknowledge your letters personally and still be able to keep the newsletter rolling along. I owe some of you thanks for passing the first letter along to a friend. I am beginning to see some subscriptions from people who saw someone else's first letter!

I received a letter from John Craig of Creative Computing yesterday, saying that he will "definitely try to get a plug in for the FLEX User's Group". I'm not quite sure just exactly what that means, guess it's something like Eisenhower's statement that the budget was tentatively finalized!!

Further note on Hemenway Associates is that there are still some problems with STRUBAL, full report next time, as we are still communicating over the difficulties. In the process, I managed to get Hemenway's Relocatable Macroassembler. I must report that it is very straightforward to use, very slight variations from Motorola and TSC. It produces relocatable files which, with the Linking loader, or the new Linkage Editor, may be loaded anywhere in memory, or put directly in a disk file as an absolute binary file. I was able to take one of my source files and with very minor editing, get it to assemble as a relocatable file! It is very nice.

```
                        NAM     CLRMEM

                        TTL     UTILITY TO CLEAR MEMORY

                        OPT     PAG

                    *
                    *  WRITTEN BY J.K. JORDAN
                    *             103 ELLIOT CIRCLE
                    *             OAK RIDGE, TN 37830
                    *             12/l4/78
                    *
                    * THIS PROGRAM WILL CLEAR ALL MEMORY
                    * BETWEEN TWO ADDRESSES (INCLUSIVE).
                    * THE COMMAND FORMAT IS:
                    *   +++CLRMEM,§START†,§END†
                    * WITH ADDRESSES IN HEXIDECIMAL.
                    * OPTIONALLY A FILL BYTE MAY BE SPECIFIED:
                    *   +++CLRMEM,§START†,§END†,§BYTE†
                   * FILLING THE MEMORY WITH THE 8 BIT
                    * VALUE OF §BYTE†.
                    *
                    *
                    * EQUATES FOR MINI-FLEX, LARGE FLEX AS COMMENTS
                    *
7103                WARMS   EQU     $7103     $AD03
7112                PUTCHR  EQU     $7112     $AD18
7118                PSTRNG  EQU     $7118     $AD1E
711E                PCRLF   EQU     $711E     $AD24
7139                OUTHEX  EQU     $7139     $AD3C
713F                GETHEX  EQU     $713F     $AD42
                    *
                    * USERS OF LARGE FLEX INSERT OUTADR EQU $AD45 AND
                    * DELETE THE OUTADR SUBROUTINE AT THE END OF THIS
                    * LISTING. MINIFLEX DOES NOT HAVE THIS ROUTINE.
                    *
7600                        ORG     $7600     $A100 FOR LARGE FLEX

7600 20 01          CLRMEM  BRA     CLR
7602 01             VER     FCB     1
7603 BD 71 3F       CLR     JSR     GETHEX    GET START ADDRESS
7606 24 09                  BCC     C1
                    *
7608 CE 76 9A       ERROR   LDX     #ERMSG    PRINT ERROR MESSAGE
760B BD 71 18               JSR     PSTRNG
760E 7E 71 03               JMP     WARMS     TO DOS
                    *
```

```
7611 FF 76 94  C1      STX    ADDR      SAVE START ADDRESS
7614 BD 7l 3F          JSR    GETHEX    GET END ADDRESS
7617 25 EF             BCS    ERROR
7619 FF 76 98          STX    ENDADR    SAVE END ADDRESS
761C BD 71 3F          JSR    GETHEX    GET FILL BYTE (OR ZERO)
761F 25 E7             BCS    ERROR
7621 FF 76 96          STX    BFILL     SAVE FILL BYTE
7624 BD 71 1E          JSR    PCRLF
7627 F6 76 97          LDA B  BFILL+1   SET FILL BYTE
762A FE 76 94          LDX    ADDR      SET START ADDRESS
762D 09                DEX
                       *
                       * MAIN LOOP
                       *
762E 08        LOOP    INX
762F E7 00             STA B  0,X       SET MENORY
7631 A6 00             LDA A  0,X       CHECK MEMORY
7633 11                CBA
7634 26 14             BNE    END1      EARLY END
7636 8C 76 2D          CPX    #LOOP-1
7639 27 07             BEQ    END2      ATTEMPT TO WIPEOUT LOOP
763B BC 76 98          CPX    ENDADR
763E 26 EE             BNE    LOOP      DO TILL END
7640 20 12             BRA    END3      NORMAL END
                       *
7642 FF 76 98  END2    STX    ENDADR    SAVE ACTUAL END ADDRESS
7645 CE 76 AB          LDX    #WIPOUT   REPORT WIPOUT
7648 20 07             BRA    END1A
764A 09        END1    DEX
764B FF 76 98          STX    ENDADR    SAVE ACTUAL END ADDRESS
764E CE 76 BF          LDX    #ENDMSG   REPORT EARLY END
7651 BD 71 18  END1A   JSR    PSTRNG
7654 5D        END3    TST B            MEM. CLEARED OR FILLED?
7655 26 08             BNE    FILLED
7657 CE 76 D8          LDX    #CLRMSG   PRINT "CLEARED STRING
765A BD 71 18          JSR    PSTRNG
765D 20 11             BRA    C2
765F CE 76 F0  FILLED  LDX    #FILMSG   PRINT "FILLED" STRING
7662 BD 71 18          JSR    PSTRNG
7665 CE 76 97          LDX    #BFILL+1  PRINT FILL BYTE
7668 BD 71 39          JSR    OUTHEX
766B CE 76 E4          LDX    #FRMSG    PRINT "FROM "(WITHOUT CR/LF)
766E 8D 17             BSR    PSTR
7670 CE 76 94  C2      LDX    #ADDR
7673 BD 77 04          JSR    OUTADR    PRINT START ADDRESS
7676 CE 76 EB          LDX    #TOMSG    PRINT " TO "(WITHOUT CR/LF)
7679 8D 0C             BSR    PSTR
767B CE 76 98          LDX    #ENDADR   PRINT END ADDRESS
767E BD 77 04          JSR    OUTADR
7681 BD 71 1E          JSR    PCRLF
7684 7E 71 03          JMP    WARMS     TO DOS
                       *
7687 A6 00     PSTR    LDA A  0,X       ROUTINE TO PRINT A
7689 81 04             CMP A  #4        CHARACTER STRING WITHOUT
```

```
768B 26 01              BNE     P1          FIRST OUTPUTTING A CARRIAGE
768D 39                 RTS                 RETURN AND LINE FEED.
768E BD 71 12   P1      JSR     PUTCHR
7691 08                 INX
7692 20 F3              BRA     PSTR
                *
7694            ADDR    RMB     2
7696            BFILL   RMB     2
7698            ENDADR  RMB     2
                *
769A 48         ERMSG   FCC     /HEX INPUT ERROR?
769B 45 58
769D 20 49
769F 4E 50
76A1 55 54
76A3 20 45
76A5 52 52
76A7 4F 52
76A9 3F
76AA 04                 FCB     4
                *
76AB  57        WIPOUT  FCC     /WIPED OUT PROGRAM - /
76AC  49 50
76AE  45 44
76B0  20 4F
76B2  55 54
76B4  20 50
76B6  52 4F
76B8  47 52
76BA  41 4D
76BC  20 2D
76BE  20
76BF  45        ENDMSG  FCC     /END ADDRESS NOT REACHED./
76C0  4E 44
76C2  20 41
76C4  44 44
76C6  52 45
76C8  53 53
76CA  20 4E
76CC  4F 54
76CE  20 52
76D0  45 41
76D2  43 48
76D4  45 44
76D6  2E
76D7  04                FCB     4
                *
76D8 4D         CLRMSG  FCC     /MEMORY CLEAR/
76D9 45 4D
76DB 4F 52
76DD 59 20
76DF 43 4C
76E1 45 41
76E3 52
```

```
 76E4 20        FRMSG   FCC    / FROM /
 76E5 46 52
 76E7 4F 4D
 76E9 20
 76EA 04                FCB    4
                *
 76EB 20        TOMSG   FCC    / TO /
 76EC 54 4F
 76EE 20
 76EF 04                FCB    4
                *
 76F0 4D        FILMSG  FCC    /MEMORY FILLED WITH /
 76F1 45 4D
 76F3 4F 52
 76F5 59 20
 76F7 46 49
 76F9 4C 4C
 76FB 45 44
 76FD 20 57
 76FF 49 54
 7701 48 20
 7703 04                FCB    4
                *
 7704 BD 71 39  OUTADR  JSR    OUTHEX
 7707 08                INX
 7708 BD 71 39          JSR    OUTHEX
 770B 39                RTS
                *
                        END    CLRMEM    SET TRANSFER ADDRESS
```

NO ERROR(S) DETECTED

SYMBOL TABLE:

ADDR	7694	BFILL	7696	C1	7611	C2	7670	CLR	7603
CLRMEM	7600	CLRMSG	76D8	END1	764A	END1A	7651	END2	7642
END3	7654	ENDADR	7698	ENDMSG	76BF	ERMSG	769A	ERROR	7608
FILLED	765F	FILMSG	76F0	FRMSG	76E4	GETHEX	713F	LOOP	762E
OUTADR	7704	OUTHEX	7139	P1	768E	PCRLF	711E	PSTR	7687
PSTRNG	7118	PUTCHR	7112	TOMSG	76EB	VER	7602	WARMS	7103
WIPOUT	76AB								

```
                            NAM    MAP

                            TTL    MAP OF FILLED MEMORY

                            OPT    PAG

                       *
                       * THIS PROGRAM WILL PRINT OUT THE
                       * START AND END ADDRESSES OF ALL
                       * NON-ZERO BLOCKS OF MEMORY SEPARATED
                       * BY AT LEAST 5 ZERO BYTES
                       *     J.K. JORDAN
                       *     103 ELLIOT CIRCLE
                       *     OAK RIDGE,TN 37830
                       *     12/14/78
                       *
                       * FORMAT:   +++MAP,§START†,§END†
                       *
                       * EQUATES FOR MINIFLEX WITH LARGE FLEX AS COMMENTS
                       *
713F                   GETHEX  EQU    $713F     $AD42
711E                   PCRLF   EQU    $711E     $AD24
7118                   PSTRNG  EQU    $7118     $AD1E
7112                   PUTCHR  EQU    $7112     $AD18
7103                   WARMS   EQU    $7103     $AD03
7139                   OUTHEX  EQU    $7139     $AD3C
                       *
7600                           ORG    $7600     $A100
                       *
7600 20 01             MAP     BRA    MAP1
7602 01                VER     FCB    1
7603 BD 71 3F          MAP1    JSR    GETHEX    GET START ADDRESS
7606 25 08                     BCS    ERROR
7608 FF 76 9B                  STX    ADDR      SAVE START ADDRESS
760B BD 71 3F                  JSR    GETHEX    GET END ADDRESS
760E 24 09                     BCC    M1
7610 CE 76 Al          ERROR   LDX    #ERMSG    PRINT ERROR MESSAGE
7613 BD 71 18                  JSR    PSTRNG
7616 7E 71 03                  JMP    WARMS     TO DOS
                       *
7619 08                M1      INX
761A FF 76 9D                  STX    END       SAVE END ADDRESS+1
761D BD 71 1E                  JSR    PCRLF
7620 CE 76 BB                  LDX    #STMSG    PRINT START ADDRESS
7623 BD 71 18                  JSR    PSTRNG
7626 CE 76 9B                  LDX    #ADDR
7629 BD 76 CE                  JSR    OUTADR
```

```
762C BD 71 1E          JSR    PCRLF
762F BD 71 1E          JSR    PCRLF
7632 FE 76 9B          LDX    ADDR
7635 09                DEX
7636 08        FINDNZ  INX             FIND 1ST NON-ZERO BYTE
7637 BC 76 9D          CPX    END
763A 26 19             BNE    M2
763C CE 76 C6          LDX    #ENDMSG  PRINT END MESSAGE
763F BD 71 18          JSR    PSTRNG
7642 FE 76 9D          LDX    END
7645 09                DEX
7646 FF 76 9D          STX    END
7649 CE 76 9D          LDX    #END
764C BD 76 CE          JSR    OUTADR
764F BD 71 1E          JSR    PCRLF
7652 7E 71 03          JMP    WARMS
                *
7655 6D 00     M2      TST    0,X
7657 27 DD             BEQ    FINDNZ
7659 FF 76 9B          STX    ADDR
765C CE 76 9B          LDX    #ADDR
765F BD 76 CE          JSR    OUTADR   PRINT START OF BLOCK
7662 86 2D             LDA A  #'-
7664 BD 71 12          JSR    PUTCHR
7667 FE 76 9B          LDX    ADDR
766A FF 76 9F  FINDZ   STX    TMPEND   FIND NEXT ZERO BYTE
766D 08                INX
766E BC 76 9D          CPX    END
7671 26 12             BNE    M3
7673 09        PEND    DEX
7674 FF 76 9B          STX    ADDR
7677 CE 76 9F          LDX    #TMPEND
767A BD 76 CE          JSR    OUTADR   PRINT END OF BLOCK ADDR.
767D BD 71 1E          JSR    PCRLF
7680 FE 76 9B          LDX    ADDR
7683 20 B1             BRA    FINDNZ   GO FIND NEXT
7685 6D 00     M3      TST    0,X
7687 26 E1             BNE    FINDZ
7689 C6 04             LDA B  #4
768B 08        CHECK4  INX             CHECK 4 MORE BYTES FOR ZERO
768C BC 76 9D          CPX    END
768F 27 E2             BEQ    PEND     (ZERO TO END)
7691 5D                TST B
7692 27 DF             BEQ    PEND
7694 5A                DEC B
7695 6D 00             TST    0,X
7697 27 F2             BEQ    CHECK4
7699 20 CF             BRA    FINDZ
                *
769B           ADDR    RMB    2
769D           END     RMB    2
769F           TMPEND  RMB    2
                *
76A1 48        ERMSG   FCC    /HEX ADDRESS INPUT ERROR/
```

```
 76A2 45 58
 76A4 20 41
 76A6 44 44
 76A8 52 45
 76AA 53 53
 76AC 20 49
 76AE 4E 50
 76BO 55 54
 76B2 20 45
 76B4 52 52
 76B6 4F 52
 76B8 0D                 FCB   $D,$A,4
 76B9 0A 04
 76BB 53        STMSG    FCC   /START AT /
 76BC 54 41
 76BE 52 54
 76C0 20 41
 76C2 54 20
 76C4 00 04              FDB   4
 76C6 45        ENDMSG   FCC   /END AT /
 76C7 4E 44
 76C9 20 41
 76CB 54 20
 76CD 04                 FCB   4
                *
 76CE BD 71 39  OUTADR   JSR   OUTHEX
 76D1 08                 INX
 76D2 BD 71 39           JSR   OUTHEX
 76D5 39                 RTS
                *
                         END   MAP
```

NO ERROR(S) DETECTED

SYMBOL TABLE:

ADDR	769B	CHECK4	768B	END	769D	ENDMSG	76C6	ERMSG	76A1
ERROR	7610	FINDNZ	7636	FINDZ	766A	GETHEX	713F	M1	7619
M2	7655	M3	7685	MAP	7600	MAP1	7603	OUTADR	76CE
OUTHEX	7139	PCRLF	711E	PEND	7673	PSTRNG	7118	PUTCHR	7112
STMSG	76BB	TMPEND	769F	VER	7602	WARMS	7103		

```
                        NAM    DUMP

                        TTL    HEX ASCII DUMP WITH OPTIONS

                        OPT    PAG
                *
                * WRITTEN BY JOHN K. JORDAN
                *            103 ELLIOT CIRCLE
                *            OAK RIDGE TN 37830
                *            12/20/78
                *
                * THIS PROGRAM WILL DUMP MEMORY TO A
                * TERMINAL IN THE FORM OF ASCII CHARS.
                * OR HEXIDECIMAL NUMBERS OR BOTH.
                * THE COMMAND IS IN THE FORM OF:
                *
                *    +++DUMP,§START†,§END†,[+§OPTIONS†l
                * WHERE [+§OPTIONS†l IS OPTIONAL.
                *
                * WHEN SPECIFIED, SOPTIONST MAY BE:
                *     A - SUPPRESS ASCII CHR. PRINTING
                *     H - SUPPRESS HEX PRINTING
                *     S - SUPPRESS EXTRA SPACES BETWEEN HEX BYTES.
                * IF NO OPTIONS ARE SPCEIFIED, HEX AND
                * ASCII WILL BE PRINTED WITH SPACES.
                *
                * EQUATES FOR MINIFLEX WITH LARGE FLEX AS COMMENTS
                *
711E            PCRLF   EQU    $711E     $AD24
713F            GETHEX  EQU    $713F     $AD42
7118            PSTRNG  EQU    $7118     $AD1E
7103            WARMS   EQU    $7103     $AD03
7121            NXTCH   EQU    $7121     $AD27
7082            EOL     EQU    $7082     $AC02
7112            PUTCHR  EQU    $7112     $AD18
7139            OUTHEX  EQU    $7139     $AD3C
                *
                * NOTE LARGE FLEX USERS ADD OUTADR EQU $AD3C AND
                * DELETE THE OUTADR SUBROUTINE AT THE END OF THIS
                * LISTING. MINIFLEX DOES NOT HAVE THIS ROUTINE.
                *
6D00                    ORG    $6D00
                *
                * NOTE LARGE FLEX USERS MAY ORG AT $A100.
                * THE MINIFLEX UTILITY AREA AT $7600 IS NOT
                * LARGE ENOUGH FOR THIS UTILITY.  YOU MAY ORG IT
```

```
                * NEAR THE TOP OF YOUR MEMORY AS I HAVE HERE OR
                * PUT IT LOW.
                *
6D00 20 01      DUMP    BRA     DUMP1
6D02 01         VER     FCB     1
6D03 BD 71 1E   DUMP1   JSR     PCRLF
6D06 86 FF              LDA A   #$FF       SET FLAGS TO DEFAULT
6D08 B7 6E 57           STA A   PLUSF      0-ENCOUNTERED'+' IN BUFFER
6D0B B7 6E 54           STA A   ASCF       0-NO PRINT;FF-PRINT ASCII
6D0E B7 6E 55           STA A   HEXF       SAME FOR HEX
6D11 B7 6E 56           STA A   SPCF       0-NO SPACES BETWEEN BYTES
6D14 BD 71 3F           JSR     GETHEX     GET START ADDRESS
6D17 24 09              BCC     DUMP2
6D19 CE 6E 5E   ERROR   LDX     #ERMSG     REPORT ERROR
6D1C BD 71 18   PRTRTN  JSR     PSTRNG
6D1F 7E 71 03           JMP     WARMS      TO DOS
6D22 FF 6E 5A   DUMP2   STX     POINT
6D25 BD 71 3F           JSR     GETHEX     TET END ADDRESS
6D28 25 EF              BCS     ERROR
6D2A 08                 INX
6D2B FF 6E 58           STX     ENDADR     SAVE END+1
                *
                * GET ANY OPTIONS
                *
6D2E BD 71 21   OPTN    JSR     NXTCH
6D31 81 0D              CMP A   #$0D
6D33 27 42              BEQ     HEADR
6D35 B1 70 82           CMP A   EOL
6D38 27 3D              BEQ     HEADR
6D3A 7D 6C 57           TST     PLUSF
6D3D 27 0E              BEQ     OPT1
6D3F 81 2B              CMP A   #'+'
6D41 27 05              BEQ     OPT2
6D43 CE 6E 7B           LDX     #OPTMSG    'OPTION INPUT ERROR'
6D46 20 D4              BRA     PRTRTN     PRINT STRING, THEN DOS
6D48 7F 6E 57   OPT2    CLR     PLUSF
6D4B 20 El              BRA     OPTN       GET NEXT CHARACTER
6D4D 81 53      OPT1    CMP A   #'S
6D4F 26 05              BNE     OPT3
6D51 7F 6E 56           CLR     SPCF
6D54 20 D8              BRA     OPTN
6D56 81 48      OPT3    CMP A   #'H
6D58 26 0F              BNE     OPT4
6D5A 7D 6E 54           TST     ASCF
6D5D 26 05              BNE     OPT5
6D5F CE 6E 8E   OPERR   LDX     #AHMSG     'BOTH A AND H ...'
6D62 20 B8              BRA     PRTRTN
6D64 7F 6E 55   OPT5    CLR     HEXF
6D67 20 C5              BRA     OPTN
6D69 81 41      OPT4    CMP A   #'A
6D6B 26 Cl              BNE     OPTN
6D6D 7D 6E 55           TST     HEXF
6D70 27 ED              BEQ     OPERR
6D72 7F 6E 54           CLR     ASCF
```

```
6D75 20 B7              BRA    OPTN
                *
                * PRINT HEADER
                *
6D77 CE 6E 76  HEADR    LDX    #HDMSG
6D7A BD 71 18           JSR    PSTRNG
6D7D 8D 37              BSR    OUT3S
6D7F 86 30              LDA A  #'0        PRINT '0' FIRST
6D81 36        HEADR1   PSH A
6D82 BD 71 12           JSR    PUTCHR
6D85 32                 PUL A
6D86 81 46              CMP A  #'F        REACHED 'F'?
6D88 27 38              BEQ    PRTDMP     GO PRINT DUMP
6D8A 7D 6E 56           TST    SPCF
6D8D 27 02              BEQ    HEADR2
6D8F 8D 29              BSR    OUTSP
6D91 8D 27     HEADR2   BSR    OUTSP
6D93 4C                 INC A
6D94 81 3A              CMP A  #$3A       PAST '9'?
6D96 26 E9              BNE    HEADR1
6D98 86 41              LDA A  #'A
6D9A 20 E5              BRA    HEADR1
                *
                * SUBROUTINE TO ALIGN PRINTING WITH HEADER
                *
6D9C E6 01     ALIGN    LDA B  1,X
6D9E C4 0F              AND B  #$0F       USE LEAST SIG. 4 BITS
6DA0 Fl 6E 53  AL       CMP B  POS
6DA3 26 01              BNE    AL1
6DA5 39                 RTS
6DA6 7C 6E 53  AL1      INC    POS
6DA9 7D 6E 56           TST    SPCF
6DAC 27 02              BEQ    AL2
6DAE 8D 0A              BSR    OUTSP
6DB0 8D 06     AL2      BSR    OUT2S
6DB2 20 EC              BRA    AL
                *
                * SUBROUTINE TO PRINT SPACES
                *
6DB4 8D 00     OUT6S    BSR    OUT3S
6DB6 8D 02     OUT3S    BSR    OUTSP
6DB8 8D 00     OUT2S    BSR    OUTSP
6DBA 36        OUTSP    PSH A
6DBB 86 20              LDA A  #$20
6DBD BD 71 12           JSR    PUTCHR
6DC0 32                 PUL A
6DC1 39                 RTS
                *
                * PRINT DUMP
                *
6DC2 BD 71 1E  PRTDMP   JSR    PCRLF
6DC5 FE 6E 5A           LDX    POINT      SET START ADDRESS
6DC8 FF 6E 5C           STX    APOINT
6DCB BD 71 1E  PRLOOP   JSR    PCRLF
```

```
6DCE CE 6E 5A          LDX    #POINT
6DD1 BD 6E B6          JSR    OUTADR    PRINT ADDRESS
6DD4 8D E2             BSR    OUT2S
6DD6 7F 6E 53          CLR    POS       TST HEXF PRINT ANY HEX?
6DD9 27 34             BEQ    PRTASC    NO
6DDB CE 6E 5A          LDX    #POINT
6DDE 8D BC             BSR    ALIGN
6DE0 FE 6E 5A          LDX    POINT
6DE3 BD 71 39  HPRT    JSR    OUTHEX
6DE6 7D 6E 56          TST    SPCF
6DE9 27 02             BEQ    HPR1
6DEB 8D CD             BSR    OUTSP
6DED 08        HPR1    INX               BUMP POINTERS
6DEE 5C                INC B             (B=POS FROM ALIGN)
6DEF BC 6E 58          CPX    ENDADR    END OF DUMP?
6DF2 26 07             BNE    HPR2
6DF4 7D 6E 54          TST    ASCF
6DF7 26 09             BNE    HPR3
6DF9 20 4F             BRA    PRTEND
6DFB C1 10     HPR2    CMP B  #$10      END OF LINE?
6DFD 26 E4             BNE    HPRT
6DFF FF 6E 5A          STX    POINT     SAVE NEXT ADDRESS
6E02 BD 71 1E  HPR3    JSR    PCRLF
6E05 7D 6E 54          TST    ASCF      PRINT ANY ASCII?
6E08 27 C1             BEQ    PRLOOP    NO
6E0A 8D A8             BSR    OUT6S
6E0C 7F 6E 53          CLR    POS
6E0F CE 6E 5C  PRTASC  LDX    #APOINT
6E12 8D 88             BSR    ALIGN
6E14 FE 6E 5C          LDX    APOINT
6E17 8D A1     APRT    BSR    OUTSP
6E19 A6 00             LDA A  0,X
6E1B 84 7F             AND A  #$7F      MASK PARITY
6E1D 81 1F             CMP A  #$1F
6E1F 23 06             BLS    ASPC
6E21 81 5F             CMP A  #$5F
6E23 22 02             BHI    ASPC
6E25 20 02             BRA    APR1
6E27 86 20     ASPC    LDA A  #$20      SET FOR SPACE
6E29 BD 71 12  APR1    JSR    PUTCHR
6E2C 7D 6E 56          TST    SPCF
6E2F 27 02             BEQ    APR2
6E31 8D 87             BSR    OUTSP
6E33 08        APR2    INX               BUMP POINTERS
6E34 5C                INC B
6E35 BC 6E 58          CPX    ENDADR    END OF DUMP?
6E38 27 10             BEQ    PRTEND    YES
6E3A C1 10             CMP B  #$10      END OF LINE
6E3C 26 D9             BNE    APRT      NO
6E3E FF 6E 5C          STX    APOINT
6E41 FF 6E 5A          STX    POINT
6E44 BD 71 1E          JSR    PCRLF
6E47 7E 6D CB          JMP    PRLOOP
6E4A BD 71 1E  PRTEND  JSR    PCRLF
```

```
6E4D BD 71 1E          JSR    PCRLF
6E50 7E 71 03          JMP    WARMS
                       *
6E53           POS     RMB    1          POSITION COUNTER
6E54           ASCF    RMB    1          FLAGS
6E55           HEXF    RMB    1
6E56           SPCF    RMB    1
6E57           PLUSF   RMB    1
6ES8           ENDADR  RMB    2          FOR END ADDRESS
6E5A           POINT   RMB    2          HEX POINTER
6E5C           APOINT  RMB    2          ASCII POINTER
                       *
6E5E 48        ERMSG   FCC    /HEX ADDRESS INPUT ERROR/
6E5F 45 58
6E61 20 41
6E63 44 44
6E65 52 45
6E67 53 53
6E69 20 49
6E6B 4E 50
6E6D 55 54
6E6F 20 45
6E71 52 52
6E73 4F 52
6E75 04                FCB    4
6E76 44        HDMSG   FCC    /DUMP/
6E77 55 4D
6E79 50
6E7A 04                FCB    4
6E7B 4F        OPTMSG  FCC    /OPTION INPUT ERROR/
6E7C 50 54
6E7E 49 4F
6E80 4E 20
6E82 49 4E
6E84 50 55
6E86 54 20
6E88 45 52
6E8A 52 4F
6E8C 52
6E8D 04                FCB    4
6E8E 42        AHMSG   FCC    /BOTH 'A' AND 'H' MUST NOT /
6E8F 4F 54
6E91 48 20
6E93 27 41
6E95 27 20
6E97 41 4E
6E99 44 20
6E9B 27 48
6E9D 27 20
6E9F 4D 55
6EA1 53 54
6EA3 20 4E
6EA5 4F 54
6EA7 20
```

```
 6EA8 42                FCC     /BE SPECIFIED./
 6EA9 45 20
 6EAB 53 50
 6EAD 45 43
 6EAF 49 46
 6EB1 49 45
 6EB3 44 2E
 6EB5 04                FCB     4
                  *
 6EB6 BD 71 39  OUTADR  JSR     OUTHEX
 6EB9 08                INX
 6EBA BD 71 39          JSR     OUTHEX
 6EBD 39                RTS
                        END     DUMP
```

NO ERROR(S) DETECTED

SYMBOL TABLE:

AHMSG	6E8E	AL	6DA0	AL1	6DA6	AL2	6DB0	ALIGN	6D9C
APOINT	6E5C	APR1	6E29	APR2	6E33	APRT	6E17	ASCF	6E54
ASPC	6E27	DUMP	6D00	DUMP1	6D03	DUMP2	6D22	ENDADR	6E58
EOL	7082	ERMSG	6E5E	ERROR	6D19	GETHEX	713F	HDMSG	6E76
HEADR	6D77	HEADR1	6D81	HEADR2	6D91	HEXF	6E55	HPR1	6DED
HPR2	6DFB	HPR3	6E02	HPRT	6DE3	NXTCH	7121	OPERR	6D5F
OPT1	6D4D	OPT2	6D48	OPT3	6D56	OPT4	6D69	OPT5	6D64
OPTMSG	6E7B	OPTN	6D2E	OUT2S	6DB8	OUT3S	6DB6	OUT6S	6DB4
OUTADR	6EB6	OUTHEX	7139	OUTSP	6DBA	PCRLF	711E	PLUSF	6E57
POINT	6E5A	POS	6E53	PRLOOP	6DCB	PRTASC	6E0F	PRTDMP	6DC2
PRTEND	6E4A	PRTRTN	6D1C	PSTRNG	7118	PUTCHR	7112	SPCF	6E56
VER	6D02	WARMS	7103						

```
                        NAM     RETITLE

                        OPT     PAG
                    *
                    * JAMES L. MCVAY
                    * 12 JAN 1979
                    *
                    * THIS PROGRAM DELETES A FILE ON WORKING DRIVE
                    * NAMED TITLE.TXT AND THEN JUMPS TO PROGRAM
                    * 'BUILD' ON SYSTEM DISK TO BUILD A FILE BY
                    * THE SAME NAME.
7740                FCB     EQU     $7740
7094                BUFFPT  EQU     $7094
7127                GETFIL  EQU     $7127
7806                FMS     EQU     $7806
7142                DOCMD   EQU     $7142
7103                WARMS   EQU     $7103
713C                RPTERR  EQU     $713C
7803                FMSCLS  EQU     $7803
7118                PSTRNG  EQU     $7118
7112                PUTCHR  EQU     $7112
                    *
7600                        ORG     $7600
                    *
7600 20 03          START   BRA     BEGIN
7602 01             VS      FCB     1
7603                SAVEX   RMB     2
                    *
7605 FE 70 94       BEGIN   LDX     BUFFPT
7608 FF 76 03               STX     SAVEX
760B CE 76 6D               LDX     #TITLE
760E FF 70 94               STX     BUFFPT
7611 CE 77 40               LDX     #FCB
7614 BD 71 27               JSR     GETFIL
7617 86 0C                  LDAA    #$C
7619 CE 77 40               LDX     #FCB
761C A7 00                  STAA    0,X
761E BD 78 06               JSR     FMS
7621 26 18                  BNE     ERROR2
7623 CE 76 67               LDX     #BUILD
7626 FF 70 94               STX     BUFFPT
7629 CE 77 40               LDX     #FCB
762C BD 71 42               JSR     DOCMD
762F 5D                     TSTB
7630 26 23                  BNE     ERROR3
7632 FE 76 03               LDX     SAVEX
7635 FF 70 94               STX     BUFFPT
```

```
 7638 7E 71 03          JMP    WARMS
                *
 763B CE 77 40  ERROR2  LDX    #FCB
 763E A6 01             LDA A  1,X        GET ERROR
 7640 81 04             CMP A  #4         NO FILE?
 7642 27 09             BEQ    ERR2A
 7644 BD 71 3C          JSR    RPTERR
 7647 BD 78 03  RECVR   JSR    FMSCLS
 764A 7E 71 03          JMP    WARMS
 764D CE 76 77  ERR2A   LDX    #NOFIL
 7650 BD 71 18          JSR    PSTRNG
 7653 20 F2             BRA    RECVR
                *
 7655 FE 76 03  ERROR3  LDX    SAVEX
 7658 FF 70 94          STX    BUFFPT
 765B CE 76 84          LDX    #MSG
 765E BD 71 18          JSR    PSTRNG
 7661 17                TBA
 7662 BD 71 12          JSR    PUTCHR
 7665 20 E0             BRA    RECVR
                *
 7667 42        BUILD   FCC    /BUILD /
 7668 55 49
 766A 4C 44
 766C 20
 766D 54        TITLE   FCC    /TITLE.TXT/
 766E 49 54
 7670 4C 45
 7672 2E 54
 7674 58 54
 7676 0D                FCB    $D
                *
 7677 4E        NOFIL   FCC    /NO SUCH FILE/
 7678 4F 20
 767A 53 55
 767C 43 48
 767E 20 46
 7680 49 4C
 7682 45
 7683 04                FCB    4
                *
 7684 45        MSG     FCC    /ERROR
 7685 52 52
 7687 4F 52
 7689 20
 768A 04                FCB    4

                        END    START
```

NO ERROR(S) DETECTED

```
                            NAM   CAT2
                    *
                            OPT   PAG,NOG

                    *JAMES L. MCVAY
                    * 14 JAN 1979
                    *
                    * SEE CAT1 30 DEC 78 AND CAT2 8 JAN 79
                    * THIS AUTOMATES THE INSERTION OF DRIVE NUMBER
                    * AND TITLEING OF THE CATALOG.
                    * ALSO ALOWS SELECTION OF OUTPUT DEVICE.
                    *
 7806               FMS     EQU   $7806
 7803               FMSCLS  EQU   $7803
 7740               FCB     EQU   $7740
 713C               RPTERR  EQU   $713C
 711B               CLASS   EQU   $711B
 7118               PSTRNG  EQU   $7118
 7103               WARMS   EQU   $7103
 711E               PCRLF   EQU   $711E
 709A               PRECHR  EQU   $709A
 708C               WORKDV  EQU   $708C
 7121               NXTCH   EQU   $7121
 7091               TERM    EQU   $7091
 7094               BUFFPT  EQU   $7094
 7142               DOCMD   EQU   $7142
 7127               GETFIL  EQU   $7127
 70A3               SWITCH  EQU   $70A3
 6F00               OUTPRT  EQU   $6F00     PUT YOUR PRINT ROUTINE ENTRY
HERE

                            OPT   NOG
 0800                       ORG   $0800

 0800 20 0A         START   BRA   BEGIN2
 0802 03            VS      FCB   3

 0803               WRITPT  RMB   2
 0805               NAMEPT  RMB   2
 0807               SAVEX   RMB   2
 0809               COUT11  RMB   1
 080A               DRIVE   RMB   1
 080B 00            PRINTT  FCB   0

 080C B6 70 9A      BEGIN2  LDA A PRECHR
 080F 81 0D                 CMP A #$D       IS IT A CR?
 0811 27 18                 BEQ   WORK
 0813 81 20                 CMP A #$20
```

```
0815 26 2A            BNE    SYNERR
0817 BD 71 21         JSR    NXTCH      GET DRIVE NUMBER
081A 81 50            CMP A  #'P
081C 26 05            BNE    xx
081E B7 08 0B         STA A  PRINTT
0821 20 08            BRA    WORK
0823 80 30      XX    SUB A  #$30
0825 2B 04            BMI    WORK       LESS THAN 0?
0827 81 04            CMP A  #4
0829 2D 03            BLT    WORK1      MORE THAN 3
                *
082B B6 70 8C   WORK   LDA A  WORKDV
082E B7 08 0A   WORK1  STA A  DRIVE      INSERT DR
0831 BD 71 21         JSR    NXTCH
0834 81 0D            CMP A  #$0D
0836 27 15            BEQ    BEGIN1
0838 81 50            CMP A  #'P
083A 26 05            BNE    SYNERR
083C B7 08 0B         STA A  PRINTT
083F 20 0C            BRA    BEGIN1
                *
0841 CE 09 A2   SYNERR LDX    #STRNG
0844 BD 71 18         JSR    PSTRNG
0847 BD 78 03         JSR    FMSCLS
084A 7E 71 03         JMP    WARMS
                *
084D FE 70 94   BEGIN1 LDX    BUFFPT
0850 FF 08 07         STX    SAVEX
0853 CE 09 96         LDX    #NAME+7
0856 FF 70 94         STX    BUFFPT
0859 B6 08 0A         LDA A  DRIVE
085C 8B 30            ADD A  #$30
085E A7 00            STA A  0,X
0860 CE 77 40         LDX    #FCB
0863 BD 71 27         JSR    GETFIL
0866 25 39            BCS    ERROR0
0868 CE 77 40         LDX    #FCB
086B 86 01            LDA A  #1
086D A7 00            STA A  0,X
086F BD 78 06         JSR    FMS        OPEN FILE
0872 26 24            BNE    ERROR1
0874 BD 78 03         JSR    FMSCLS
0877 BD 71 1E         JSR    PCRLF
087A B6 08 0B         LDA A  PRINTT
087D 26 05            BNE    PAT1
087F CE 09 91         LDX    #NAME+2
0882 20 03            BRA    PAT2
                *
0884 CE 09 8F   PAT1  LDX    #NAME
0887 FF 70 94   PAT2  STX    BUFFPT
088A BD 71 42         JSR    DOCMD
088D 5D               TST B
088E 26 14            BNE    ERROR2
                *
```

```
0890 FE 08 07  CONT    LDX    SAVEX
0893 FF 70 94          STX    BUFFPT
0896 20 3D             BRA    BEGIN
0898 CE 77 40  ERROR1  LDX    #FCB
089B A6 01             LDA A  1,X        GET ERROR
089D 81 04             CMP A  #4         NO SUCH FILE?
089F 27 09             BEQ    CONT1
08A1 BD 71 3C  ERROR0  JSR    RPTERR
08A4 BD 78 03  ERROR2  JSR    FMSCLS
08A7 7E 71 03          JMP    WARMS
                *
08AA BD 78 03  CONT1   JSR    FMSCLS
08AD FE 08 07          LDX    SAVEX
08B0 FF 70 94          STX    BUFFPT
08B3 B6 08 0A          LDA A  DRIVE
08B6 8B 30             ADD A  #$30
08B8 B7 09 C1          STA A  TITLE1
08BB B6 08 0B          LDA A  PRINTT
08BE 27 0C             BEQ    PAT4
08C0 CE 6F 00          LDX    #OUTPRT
08C3 FF 71 0D          STX    $710D
08C6 7F 70 A3          CLR    SWITCH
08C9 BD 71 1E          JSR    PCRLF
08CC BD 71 1E  PAT4    JSR    PCRLF
08CF CE 09 AF          LDX    #TITLE
08D2 BD 71 18          JSR    PSTRNG
                *
08D5 CE 77 40  BEGIN   LDX    #FCB
08D8 86 06             LDA A  #6
08DA A7 00             STA A  0,X
08DC B6 08 0A          LDA A  DRIVE
08DF A7 03             STA A  3,X
08E1 BD 78 06          JSR    FMS
08E4 27 09             BEQ    PRINT
                *
08E6 BD 71 3C  ERROR   JSR    RPTERR
08E9 BD 78 03          JSR    FMSCLS
08EC 7E 71 03          JMP    WARMS
08EF B6 08 0B  PRINT   LDA A  PRINTT
08F2 27 09             BEQ    PRINT1
08F4 CE 6F 00          LDX    #OUTPRT
08F7 FF 71 0D          STX    $710D
08FA 7F 70 A3          CLR    SWITCH
08FD CE 09 C3  PRINT1  LDX    #S1        SET
0900 C6 28             LDA B  #40        BUFFER
0902 86 20             LDA A  #$20       TO ALL
0904 A7 00     LOOP    STA A  0,X        SPACES
0906 08                INX
0907 5A                DEC B
0908 26 FA             BNE    LOOP
090A CE 09 C3          LDX    #S1        GET FIRST
090D 8D 12             BSR    GETNAM     NAME
090F CE 09 D1          LDX    #S2        SECOND
0912 8D 0D             BSR    GETNAM
```

```
0914 CE 09 DF          LDX    #S3        THIRD
0917 8D 08             BSR    GETNAM
0919 CE 09 C3          LDX    #S1        PRINT
091C BD 71 18          JSR    PSTRNG     THEM
091F 20 CE             BRA    PRINT

                *
0921 FF 08 03  GETNAM  STX    WRITPT     SAVE POINTER
0924 CE 77 40  GETLOP  LDX    #FCB
0927 86 07             LDA A  #7         RETRIEVE
0929 A7 00             STA A  0,X        DIRECTORY
092B BD 78 06          JSR    FMS        ENTRY
092E 26 53             BNE    ERR2
0930 CE 77 44          LDX    #FCB+4     POINT AT
0933 FF 08 05          STX    NAMEPT     DIRECTORY NAME
0936 A6 00             LDA A  0,X        GET FIRST CHR
0938 27 33             BEQ    ENDIT      IF 0
093A 2B E8             BMI    GETLOP     IF MINUS
093C C6 0B             LDA B  #11        11 CHARACTERS
093E F7 08 09          STA B  COUT11     IN NAME
0941 5F                CLR B
                *
0942 FE 08 05  LOOP1   LDX    NAMEPT     GET CHARACTER
0945 A6 00             LDA A  0,X
0947 81 00             CMP A  #0         IS IT PRINTABLE?
0949 27 08             BEQ    NOLTR      -NO-
094B FE 08 03          LDX    WRITPT     YES, STORE
094E A7 00             STA A  0,X        IN STRING
0950 7C 08 04          INC    WRITPT+1   BUMP
                *
0953 7C 08 06  NOLTR   INC    NAMEPT+1   POINTERS
0956 5C                INC B
0957 C1 08             CMP B  #8         IF LAST OF NAME
0959 27 06             BEQ    POINT      INSERT POINT
                *
095B 7A 08 09  DECOUT  DEC    COUT11     IF LAST OF NAME
095E 26 E2             BNE    LOOP1      PLUS EXTENSION
0960 39                RTS               RETURN
                *
0961 86 2E     POINT   LDA A             INSERT
0963 FE 08 03          LDX    WRITPT     EXTENSION
0966 A7 00             STA A  0,X        POINT
0968 7C 08 04          INC    WRITPT+1
096B 20 D5             BRA    LOOP1
                *
096D CE 09 C3  ENDIT   LDX    #S1        PRINT LAST
0970 BD 71 18          JSR    PSTRNG     STRING
0973 BD 71 1E          JSR    PCRLF
0976 CE 77 40          LDX    #FCB
0979 86 04             LDA A  #4         CLOSE
097B A7 00             STA A  0,X        DIRECTORY
097D BD 78 06          JSR    FMS
0980 7E 71 03          JMP    WARMS
```

```
                    *
 0983 CE 77 40  ERR2    LDX    #FCB
 0986 A6 01             LDA A  1,X        CHECK FOR
 0988 81 08             CMP A  #8         EOF
 098A 27 El             BEQ    ENDIT
 098C 7E 08 E6          JMP    ERROR
                    *
 098F 50        NAME    FCC    /P,LIST   .TITLE.TXT/
 0990 2C 4C
 0992 49 53
 0994 54 20
 0996 20 2E
 0998 54 49
 099A 54 4C
 099C 45 2E
 099E 54 58
 09A0 54
 09A1 0D                FCB    $0D
 09A2 53        STRNG   FCC    /SYNTAX ERROR/
 09A3 59 4E
 09A5 54 41
 09A7 58 20
 09A9 45 52
 09AB 52 4F
 09AD 52
 09AE 04                FCB    4
 09AF 43        TITLE   FCC    /CATALOG FOR DRIVE /
 09B0 41 54
 09B2 41 4C
 09B4 4F 47
 09B6 20 46
 09B8 4F 52
 09BA 20 44
 09BC 52 49
 09BE 56 45
 09C0 20
 09C1 00 04     TITLE1  FDB    $0004
                    *
 09C3           S1      RMB    14
 09D1           S2      RMB    14
 09DF           S3      RMB    12
 09EB 04                FCB    4

                        END    START
```

NO ERROR(S) DETECTED

```
1 REM *** FLEXSUBR
2 REM *** CALLS FLEX AS SUBR. AND RETURNS TO BASIC
4 REM *** A$ MUST BE DOS CMD STRING
5 REM ***   *** AN ORIGINAL PROGRAM BY
6 REM ***       RICHARD G CAGLE
7 REM ***       % APPLEVALLEY DAY SCHOOL
8 REM ***       3926 ERIE; HOUSTON, TX 77087
9 REM *** SWTPC VER 3.0 DISK BASIC
10 INPUT "HEY! GIVE ME YOUR DOS CMD",A$
20 GOTO 40
30 POKE(X,Y):RETURN
35 X=X+1:RETURN
40 X=28820:Y=112:GOSUB 30:GOSUB 35:Y=0:GOSUB 30
55 A=LEN(A$).B=28672:C=B+A
60 Y1=1:FOR X=B TO C
65 Y=ASC(MID$(A$,Y1)):Y1=Y1+1
70 GOSUB 30:NEXT X
75 Y=13:GOSUB30:X=103:Y=113:GOSUB 30
85 GOSUB 35:Y=66:GOSUB 30:LET A=USER(X):PRINT
90 PRINT:PRINT"DOS CMD COMPLETE":STOP
```

```
                        NAM    SEARCH
                    *
                        TTL    SEARCH FOR HEX OR ASCII STRING
                    *
                        OPT    PAG
                    * THIS PROGRAM SEARCHES ANY BLOCK OF
                    * MEMORY FOR ANY LENGTH STRING
                    * ADAPTED FROM MY SEARCH PROGRAM
                    * 73'S MAGAZINE MAR-78
                    * GARRY O CAUDELL- JAN 79
                    * 3125 ROBIN LYNN DR.
                    * ASHLAND,KY 41101
                    * PERMISSION FOR REPRODUCTION GRANTED
                    *
                    * FLEX EQUATES
                    *
7112                PUTCHR  EQU    $7112
7118                PSTRNG  EQU    $7118
7139                OUTHEX  EQU    $7139
74BA                HEXADJ  EQU    $74BA
7115                INBUFF  EQU    $7115
713F                GETHEX  EQU    $713F
710F                GETCHR  EQU    $710F
711E                CRLF    EQU    $711E
                    *
7600                        ORG    $7600
                    *
7600 20 01          START   BRA    START1    FLEX CONVENTION
7602 01                     FCB    $01       VERSION NUMBER
7603 BD 76 C8       START1  JSR    CLEAR     CLEAR SCREEN
7606 CE 76 E7               LDX    #MSTART   PROMPT FOR START ADDRESS
7609 BD 71 18               JSR    PSTRNG
760C BD 76 C1               JSR    BADDR     GET START ADDRESS
760F FF 77 18               STX    ASTART    STORE IT
7612 BD 71 1E               JSR    CRLF
7615 CE 76 EE               LDX    #MSTOP    STOP ADDRESS
7618 BD 71 18               JSR    PSTRNG
761B BD 76 C1               JSR    BADDR     GET STOP ADDRESS
761E FF 77 1A               STX    ASTOP     STORE IT
                    *
                    * ASCII OR HEX?
                    *
7621 BD 71 1E       REPEAT  JSR    CRLF
7624 CE 77 00               LDX    #MASCII   ASCII OR HEX?
7627 BD 71 18               JSR    PSTRNG
762A BD 71 0F               JSR    GETCHR
762D 81 41                  CMP A  #$41      IS IT "A"
```

```
762F 26 1B              BNE    HEX
                 *
                 * BUILD  ASCII STRING
                 *
7631 BD 71 1E            JSR    CRLF
7634 CE 76 F4            LDX    #MSEARCH  SEARCH FOR?
7637 BD 71 18            JSR    PSTRNG
763A CE 77 22            LDX    #ASTRING  POINT TO STRING
763D BD 71 0F  INASC     JSR    GETCHR    GET BYTE
7640 A7 00               STA A  0,X       STORE IT
7642 08                  INX
7643 81 0D               CMP A  #$0D      END?
7645 26 F6               BNE    INASC
7647 09                  DEX               CLEAR FOR FLAG
7648 6F 00               CLR    0,X
764A 20 17               BRA    SEARCH
                 *
                 * BUILD  HEX STRING
                 *
764C BD 71 1E  HEX       JSR    CRLP
764F CE 76 F4            LDX    #MSEARCH  SEARCH FOR?
7652 BD 71 18            JSR    PSTRNG
7655 CE 77 22            LDX    #ASTRING  POINT TO STRING
7658 BD 76 CD  HEX1      JSR    INHEX     GET HEX BYTE
765B A7 00               STA A  0,X       STORE IT
765D 08                  INX
765E 22 F8               BHI    HEX1      FINISHED?
7660 09                  DEX
7661 6F 00               CLR    0,X       CLEAR FLAG
                 *
                 * CHECK  FOR MATCH
                 *
7663 FE 77 18  SEARCH    LDX    ASTART    GET START ADDRESS
7666 FF 77 1C            STX    PADDR     STORE FOR PRESENT ADDRESS
7669 FE 77 1C  NEXT      LDX    PADDR     GET PRESENT ADDRESS
766C FF 77 1E            STX    PADDR1    STORE TEMP
766F CE 77 22            LDX    #ASTRING  POINT TO STRING
7672 FF 77 20            STX    STRING    STRING TEMP
7675 FE 77 1E  LOOP      LDX    PADDR1    GET TEMP
7678 A6 00               LDA A  0,X       GET BYTE
767A FE 77 20            LDX    STRING    POINT TO STRING
767D Al 00               CMP A  0,X       MATCH?
767F 26 21               BNE    TESTX     NO? SEE IF FINISHED
7681 6D 01               TST    1,X       YES? SEE IF END OF STRING
7683 27 10               BEQ    PRINT     IF STRNG MATCH PRINT ADDRESS
7685 FE 77 1E            LDX    PADDR1
7688 08                  INX               ADVANCE TEMP
7689 FF 77 1E            STX    PADDR1
768C FE 77 20            LDX    STRING    ADVANCE STRING
768F 08                  INX
7690 FF 77 20            STX    STRING
7693 20 E0               BRA    LOOP      GO BACK FOR REST OF STRING
                 *
                 * PRINT  ADDRESS
```

```
                    *
7695 BD 71 1E  PRINT   JSR    CRLF
7698 CE 77 1C          LDX    #PADDR     GET PRESENT ADDRESS
769B BD 71 39          JSR    OUTHEX     FIRST HALF
769E 08                INX
769F BD 71 39          JSR    OUTHEX     SECOND HALF
                    *
                    * SEE IF FINISHED
                    *
76A2 FE 77 1C  TESTX   LDX    PADDR      WHERE ARE WE?
76A5 08                INX               ADVANCE ONE BYTE
76A6 FF 77 1C          STX    PADDR
76A9 BC 77 1A          CPX    ASTOP
76AC 26 BB             BNE    NEXT       NOT FINISHED? GO BACK
76AE CE 77 0E          LDX    #FIN       WANT MORE?
76B1 BD 71 18          JSR    PSTRNG
76B4 BD 71 0F          JSR    GETCHR
76B7 81 59             CMP A  #$59       "Y" (YES)
76B9 27 03             BEQ    FLEX       GOTO FLEX
76BB 7E 76 21          JMP    REPEAT
76BE 7E 71 03  FLEX    JMP    $7103
                    *
                    * SUBROUTINES
                    *
76C1 BD 71 15  BADDR   JSR    INBUFF
76C4 BD 71 3F          JSR    GETHEX
76C7 39                RTS
                    *
                    * CLEAR SCREEN
                    *
76C8 86 1A     CLEAR   LDA A  #$1A
76CA 7E 71 12          JMP    PUTCHR     HIDDEN RTS
                    *
                    * INPUT HEX
                    *
76CD BD 71 0F  INHEX   JSR    GETCHR     GET BYTE
76D0 BD 74 BA          JSR    HEXADJ     CONVERT TO HEX
76D3 25 10             BCS    EREXIT     INVALID?
76D5 16        INHX1   TAB
76D6 58                ASL B             SHIFT LEFT TO HI ORDER
76D7 58                ASL B
76D8 58                ASL B
76D9 58                ASL B
76DA BD 71 0F          JSR    GETCHR     GET SECOND PART
76DD BD 74 BA          JSR    HEXADJ
76E0 25 03             BCS    EXEXIT     INVALID?
76E2 1B                ABA               ADD TO FIRST HEX DIGIT
76E3 0C                CLC               IF ALL IS WELL
76E4 39                RTS
76E5 0D        EREXIT  SEC               FLAG NONHEX
76E6 39                RTS
                    *
                    * MESSAGES
                    *
```

```
 76E7 53        MSTART  FCC    /START /
 76E8 54 41
 76EA 52 54
 76EC 20
 76ED 04                FCB    $04
                *
 76EE 53        MSTOP   FCC    /STOP /
 76EF 54 4F
 76F1 50 20
 76F3 04                FCB    $04
                *
 76F4 53        MSEARCH FCC    /SEARCH FOR /
 76F5 45 41
 76F7 52 43
 76F9 48 20
 76PB 46 4F
 76FD 52 20
 76FF 04                FCB    $04
                *
 7700 41        MASCII  FCC    /ASCII OR HEX /
 7701 53 43
 7703 49 49
 7705 20 4F
 7707 52 20
 7709 48 45
 770B 58 20
 770D 04                FCB    $04
                *
 770E 46        FIN     FCC    /FINISHED /
 770F 49 4E
 7711 49 53
 7713 48 45
 7715 44 20
 7717 04                FCB    $04
                *
                * PROGRAM EQUATES
                *
                * USED FDB'S INSTEAD OF RMB'S TO
                * ASSQRE INITIALIZATION TO ZERO
                *
 7718 00 00     ASTART  FDB    $0000
 771A 00 00     ASTOP   FDB    $0000
 771C 00 00     PADDR   FDB    $0000
 771E 00 00     PADDR1  FDB    $0000
 7720 00 00     STRING  FDB    $0000
 7722 00 00     ASTRING FDB    $0000
                        END    START
```

NO ERROR(S) DETECTED