THE CBUG ESCAPE

TABLE OF CONTENTS

CHICAGO B128 USER'S GROUP -- INTERNATIONAL (CBUG, Inc.)
4102 N. Odell, Norridge, Il. 60634, U.S.A.

THE CBUG ESCAPE

TENTH ISSUE
Spring 1988

* *

THE CBUG ESCAPE is published 4 times a year, more or less, by the Chicago B128 User's Group - International (CBUG, Inc.), an international membership organization in support of applications and usage of the Commodore B128 Computer. Note that some issues are combined in one publication.

AREA CODE CHANGE NOTICE: **Effective in November 1989 (one year from now), our phone area code will become 708, the rest of the number will remain the same.**

CHECK YOUR ADDRESS LABEL FOR **EXPIRATION** DATE. The expiration date is located at the top of the address label in the form of YYMM, eg. 8712 indicates an expiration at the end of December 1987. CBUG will be unable to mail publications without renewal. **PLEASE KEEP YOUR MAILING ADDRESS CURRENT.** Renewal invoices will be mailed early Feb. 1989.

CBUG is NOT affiliated or allied with any other organization, users' group business or other entity of any kind, except in support of CBUG chapters.

Advertisements, articles and contents of disks are solely the responsibility of the individual authors. Their existence in a CBUG publication does not imply any endorsement by CBUG. Please report to CBUG exceptional performance, either pro or con.

Publishing address: CBUG, Inc. c/o Norman Deltzke, 4102 N. Odell, Norridge, Il. 60634 USA. 312-456-8720 7pm to 11pm CST.

NOTE: Editorial additions to articles are donoted by enclosing the added text in double angle brackets: <<Example of editor added text marking.>>

Cover price this issue: $6.00. 1988 subscription rate: $14.00 (bulk rate, U.S. & possessions ONLY); $20.00 (first class, U.S. & possessions, Canada & Mexico); $21.00 (surface <boat> first class, all others); $35.00 (small packet rate air mail, any country).

* *

SCRATCH PAD

By: Norman Deltzke
Sept. 24, 1988

What can I say, late as usual! The "Summer" issue will contain a complete reprint of library listings 1985 thru 1988. You've probably noticed that many library listings are explained in articles. It is advised that you order any past issues you do not have!

CAUTION -- CAVEAT EMPTOR. We have been informed that there are two firms advertising in the popular press, B-128 computers, and 8050 drives at absolutely incredibly low prices; one even advertises them as new!! BALDERDASH -- you get what you pay for -- cheap = garbage. Generally speaking the reports received indicate used, canabalized, damaged and usually inoperable units. Purchase such stuff ONLY for spare parts value.

AUTHORS NEEDED As we approach the beginning of our 5th publishing year, we need to expand our author base. Many of our regulars have taken on new obligations, employments, etc. which limit their time. How to articles dealing with our commercial or popular library programs are amongst the most valuable to the largest number of readers. You can only expect to get from a user's group in proportion to what you put in!

Fast Bus, the joint project of Messrs. Anderson, Jarvis, and Springer is being further expanded. Next issue a companion disk will be released which will allow direct transfer of other formats of disk to and from the B128. There will be about 250 formats -- virtually everything around from MSDOS to CPM. This should complete the efforts to allow easy data transfer between virtually any 5 1/4" disk based computer and the B128 with any drive compatable with it. This also completes the disk & program access problem occuring when the co-processor is added to the B-128. You will, of course need a 1571 Drive, both Fast Bus disks and Anderson's adaptor. One caution, the serial bus can not be used with Superscript II in the presence of Liz Deal's Superscript Fix -- there is an irreconcilable program clash. Superscript II without the Fix is required to operate with Fast Bus.

Inadvertantly escaping publication this issue is a paper and disk (by Clyde Northrup) on database output/export/import. Articles on this subject are requested for the next issue since this topic is specifically germain to many Fast Bus applications. Please try to get the articles/disks in by Oct. 25, 1988 for inclusion in the next issue.

How lucky we are, thanks to Tony Goceliak and Mathew Goldstein. Consistant with their continuous stellar performances, we now have a Syntax Checker (allows easy automated location of program errors), and a super heavy weight disk editor! Such things just don't exist in the world of consumer computers! Of course, for the novice there is Petersen's tutorial, and then the next step is Swan's Basic Course. These two disks together with the Programmers Reference Guide (now available from N.W. Music) provide the materials for anyone to learn effective proficiency in basic programming.

BASIC 4.0 (Extended) Tutorial

by: Warren. D. Swan

5 MORE DATA MANIPULATION

5.1 READ, DATA and RESTORE
[Handling Data 7]

Often a program needs a lot of set up at the beginning - setting variables to starting values, opening files, dimensioning arrays, and often filling them with data. One can use the READ and DATA statement pair to assign values to variables, including arrays. The READ statement includes a list of variables of any type (real, integer and string), just like the INPUT or INPUT# statements. When the READ statement is executed, BASIC scans the DATA statements for values to assign to the variables in the READ statement. The data in the DATA statements must be in the right order for being read from the READ statement, so that real and integer variables get their numeric values and strings get their string values. Here's an example:

```
    100 read a,b,c$,f%(1,1)
    200 data 23.5, 88, alfred, 37.2
```

When the READ statement is executed, it will assign 23.5 to A, 88 to B, "alfred" to C$, and 37 to F%(1,1) - the .2 is truncated since the number is being assigned to an integer variable. Notice, too, that we didn't have to put quotes around alfred in the DATA statement. This is only necessary when the string data in the DATA statements is to contain:
(1) leading or trailing spaces,
(2) commas (,) or colons (:),
(3) any shifted characters (capitals or graphics) except the shifted characters on the top row of the main keys (!@#$% etc.), or
(4) any control characters (HOME, CLR, cursor controls, etc.)
The spaces between the items in the sample DATA statement were just for readability.

The tutorial disk explains the placement of data statements and what happens when they are "executed."

Each READ statement does NOT have to have a specifically matching DATA statement. Instead, one DATA statement may provide data for several READs, as in this loop:

```
500 for i=1 to 7: read d$(i): next
600 data"Sunday","Monday","Tuesday","Wednesday","Thursday",
"Friday","Saturday"
```

Also, one READ statement may have its data spread over several DATA statements:

```
500 read e1$, e2$, e3$: rem read error messages:
600 data "Undefined identifier
610 data "Syntax error
620 data "File not found
```

There are 2 possible errors that can occur when using READ and DATA:
(1) If the next READ variable is a numeric variable and the next DATA item is bad (a string, too large a number, contains a character not allowed in numbers, etc.) the program will stop with an error.
(2) If BASIC runs out of DATA items while scanning to fulfill a READ variable, it will stop with an "out of data" error. (This can be TRAPped. We'll learn about TRAP later.)

The tutorial disk states some good programming practice suggestions for the usage of READ & DATA. It also gives examples of typical applications of READ & DATA.

BASIC continues reading data statements to fulfill all the READs it encounters, and errors out it there is no more data. The RESTORE instruction tells BASIC to start reading DATA statements all over again:

```
100 data 10,30,50,40,20
110 read a,b,c: print a,b,c
120 restore
130 read d,e: print d,e
run
 10       30       50
 10       30
```

The RESTORE in line 120 causes the READ in line 130 to read the 10 and 30 over again into D and E. Without the RESTORE it would have read the 40 and 20 into D and E.

In BASIC 4.0+ the RESTORE statement can be given an expression, which it will evaluate and will start READing data from the first DATA statement in or after the line represented by the expression. For example:

```
100 data 10,30,50
110 data 40,20
120 rem final values:
150 data 0,60
200 read a,b: print a,b
210 restore 120
220 read c,d: print c,d
run
 10       30
 0        60
```

The RESTORE 120 in line 210 causes it to skip the 50, 40 and 20. The next READ (in line 220) starts scanning for DATA from line 120, finding the 0 and 60 in the DATA statement in line 150.

Since the line number specified in a RESTORE statement can actually be an expression, it can lead to interesting random access techniques in retrieving data from DATA statements.

5.2 User Defined Functions [Handling Data 8]

We have already learned about the various functions built into BASIC. To further supplement those functions, BASIC allows us to define functions of our own. However, our functions can only be real numeric function, not integer or string functions, although we are allowed to use any of the built-in functions to construct a user defined function.

To begin with, we notice that there are no arccosine or arcsine functions in BASIC. There is an arctangent function, which is mathematically related to these functions. We will use formulas found in Appendix G, page 120, of the gray "Commodore B Series Users Guide", to determine arccosine (inverse cosine) and arcsine (inverse sine) from arctangent:

```
    120 def fn acs(r) = PI/2 - atn(r/sqr(1-r*r))
    130 def fn asn(r) = atn(r/sqr(1-r*r))
```

DEF stands for define, and FN stands for function. The FN is always required. After the FN is the actual name of the function, which must follow the rules of any variable name. In particular, FN ACS is the same as FN AC, since only the first 2 characters of a variable name are significant. Often the space is left out between the FN and the function name. The parenthesis here do NOT indicate an array. Instead, the variable placed in these parenthesis has a special meaning for this function definition, which we will see in a minute. Again, due to printer problems, the PI here is actually representing the pi character next to the RETURN key.

A function must be defined before it can be used. So, we must execute the above lines before we can use FNACS or FNASN. Here is examples of how to use these functions:

```
    230 print "The arccosine of .5 radians is" fnacs(.5)
    270 theta = fnasn(sqr(alpha)) + fnasn(beta/2)
```

We often say that BASIC "calls" a function when it uses it - just like with subroutines. When it calls the function in line 230 above, BASIC looks up the definition of FNACS and finds out that the argument, .5, will be represented by a variable, R, in the function definition. PLEASE NOTE THAT

THIS VARIABLE R HAS NOTHING WHATSOEVER TO DO WITH THE VARIABLE R ANYWHERE ELSE IN THE PROGRAM. BASIC actually uses a different, temporary variable R to "store" the number .5. The result is as if we had used:

```
   230 print "The arccosine of .5 radians is" PI/2 -
atn(.5/sqr(1-.5*.5))
```

Everywhere that the R occurs in the definition, the .5 is put in its place. If we had also used a variable R elsewhere in the program, ITS VALUE IS UNCHANGED when using this function (unless of course we had said r=fnacs(.5)).

Line 270 calls the function FNASN twice, the first time using the square root of the value stored in the variable AL(pha), and the second time using half of the value stored in the variable BE(ta). It is as if we had typed:

```
270
theta=atn(sqr(al)/sqr(1-sqr(al)*sqr(al)))+atn(be/2/sqr(1-be/
2*be/2))
  or
270  theta=atn(sqr(al)/sqr(1-al))+atn(be/2/sqr(1-be*be/4)):
rem "simplified"
```

See how much typing the function definition saved us? Not only that, but the earlier version of line 270, using FNASN, was much easier to understand!

The variable in the function definition (R above) is called the formal argument, or the formal parameter. Commodore BASIC only allows one formal argument. The function definition may use other variables, which will refer to those variables as they actually appear elsewhere in the program. These may be real, integer or string variables. Sometimes it is convenient to define a function that does not need to have anything passed to it. The tutorial disk explains how to do this.

We can use a variable called SS and a function FN SS. This is legal, just as using SS and SS(3) are legal in the same program. Each refers to a different variable or function.

5.3 Clearing Variables and Closing Files [Handling Data 9]

What if we have just run a program that we are working on, and we decide to find out how much space is being used in bank 2? Well, we can find out how much is LEFT, by typing:

```
   ?fre(2)
```

But now we have forgotten how much space was in bank 2 to start with. We could clear out all the variables by using the NEW command, but that would wipe out the program we're working on. Or we could just use the CLR instruction:

```
   clr: ?fre(2)
```

The CLR will clear all program variables so that the printing of fre(2) will show how much memory was available in that bank to start with. Then the amount of memory in bank 2 used by the program is simply the difference between these two.

The CLR instruction takes no operands, so the CLR instruction is also the CLR statement.

The CLR statement can also be used for another handy purpose. If your program stopped because of an error, all the files will still be open. This is no tragedy with INPUT files, but files that were being WRITTEN to need to be properly finished. The CLR instruction also closes all files, thus allowing the various devices to properly finish any "files" being written.

In a future section we will learn how one program can start up another. This is called CHAINING. When BASIC loads in the new program, it does NOT clear the variables that were set up by the earlier program(s), nor does it close any files that weren't explicitly (D)CLOSEd. This can be handy, since the new program can now use all the arrays, variables, and files set up by the first program. However, sometimes this new program doesn't want the old program's variables or files. So it may start with a CLR statement to wipe out all the variables and close all the files.

5.4 Illegal Direct Commands [Programming 5]

In the very first chapter of this tutorial we learned that almost all BASIC instructions, or commands, can be used both in programs as well as from DIRECT MODE; that is, typed in without a line number. There are some exceptions caused by the fact that BASIC uses the same area of memory to temporarily store command lines that it uses to execute some instructions.

The tutorial disk explains why this is true. Attempting to use any of these instructions (including some we haven't learned yet) in direct mode causes an "illegal direct" error:

```
 DEF FN     DISPOSE     GET     GET#     INPUT     INPUT#
```

5.5 Copying, Scratching, and Renaming Files [Disk Commands 4]

A whole course could be taught on creating, modifying and maintaining disk files. But there are some basics that we will cover here so that we can "get by". The three commands that will be presented here are not found in BASIC 2.0 on older Commodore machines. Yet in actuality, these commands merely send information to the disk unit to perform the operations. The disk unit itself has the "intelligence" to copy, scratch, and rename disk files. This sets Commodore disk units apart from those used on other machines.

Recall from section 4.2 that you may use expressions in disk commands, even though this section will simplify matters by using string and numeric constants. Review section 4.2 if you've forgotten how to use variables or expressions in disk commands.

5.5.1 COPY

The COPY command copies files WITHIN A SINGLE DISK UNIT. It can copy a file
(1) from one disk to another (0 to 1 or 1 to 0), or
(2) to another file name on the same disk.
These are actually performed by the DOS in the disk unit. BASIC simply passes the command along to the disk unit. In addition, DOS 2.7 (and later DOSs) allow you to copy ALL files from one disk to another. DOS 2.5 will not allow this because of a bug in the DOS. However, due to a bug in DOS 2.7, if you*try to copy a file to a disk, and a file with the same name already exists, THE DISK UNIT WILL "HANG". It will no longer respond to any commands until it is turned off and then back on.

To copy a file from drive 0 to drive 1, type:

```
   copy "filename" to d1,"*
```

The * means to keep the same filename on drive 1. If we wanted to give it a new name, we would type:

```
   copy "filename" to d1,"newname
```

We don't have to use d0 before the TO because that is the default. If we wanted to go the other way, drive 1 to drive 0, we must type:

```
   copy "filename",d1 to d0,"name
```

(where "name" is either a new name or * to keep the same name). Remember from sections 4.1 and 4.2 that it doesn't matter which order we give the parameters to the copy command; except that the source file information must stay before the TO, and the destination file information must stay after the TO.

To copy a file to a new name on a single disk, use:

```
   copy "oldname",d1 to "newname
```

In this case, the drive number only had to be given once. We put it before the TO to indicate that the file is not on drive 0 - the default; remember? If we were copying the file to a new name on drive 0, we could use:

```
    copy "oldname" to "newname
```

For those of us lucky enough to own more than one disk unit, we have to specify which unit the disk is (disks are) on:

```
    copy u9,"oldname" to "newname
```

The u9 could have been placed on either side of the TO, on either side of either file name, as long as it is present SOMEWHERE to tell BASIC which unit to send the command to.

For those of us with DOS 2.7 or later, we can copy all files from one drive to another with

```
    copy d0 to d1
or  copy d1 to d0
```

However, no file on the destination disk may have the same name as any file on the source disk, or the disk unit will hang when it encounters the clash. It will not ruin either file though. This command may be useful for merging the contents of two disks together. The BACKUP instruction seems similar, but (1) it wipes out any files on the destination disk, and (2) it copies EVERY block - even those not being used by a file. BACKUP and COPY (all files) are not interchangeable. Each has its own purpose.

The COPY command DOES NOT AUTOMATICALLY INFORM YOU OF AN ERROR. You must use the command ?ds$ (from section 3.6) to warn you if an error occurs.

The tutorial disk explains what the computer sends to the disk unit to make it perform the copy, and also explains why the copy command is better than printing to the command channgel to do the same thing.

5.5.2 RENAME

The syntax of the RENAME command is similar to that of the copy command. The main, obvious difference is that a file is renamed on ONE disk. If the file is on drive 1, the d1 must be before the TO (d0 is not needed at all):

```
    rename "old" to "new
or  rename d1,"old" to "new
```

The tutorial disk gives further examples and information.

5.5.3 SCRATCH

This command simply requires the name of the file to scratch, and the drive (if drive 1), and the unit (if not unit 8). In this course we have not covered file name patterns. This topic belongs in another course (since it is a Commodore disk feature, not a BASIC-specific feature). Suffice it to say that file name patterns are allowed when using the SCRATCH command.

If the SCRATCH command is used in direct mode, it will always ask:

```
    are you sure?
```

Type y (unshifted) and press RETURN to scratch the files; otherwise just press RETURN to cancel the operation. If you use a file name pattern, be careful that the pattern doesn't match a file name you had forgotten about. To be certain, you may want to issue a DIRECTORY command with the same pattern first.

The tutorial disk explains how 'header "disk name",d0' is a much faster way to erase all the files on a disk than 'scratch "*"'.

Do not use a comma in the file name given to the scratch command. However, the tutorial disk explains when you can do this to scratch multiple files with one scratch command, if you are very careful.

5.6 Communication with Multiple Devices [Disk Files 4]

This subject, while extremely useful, is a little heavy for the kind of information we like to present in this abridged version of the tutorial. It is discussed fully in the tutorial disk.

5.7 Comments After Line Numbers [Comments 2]

In chapter 1 we learned about the importance of comments, some common sense commenting bylaws, and about the REM statement. The makers of Commodore BASIC know how distressed a programmer can become when reviewing someone else's program, or even reviewing one's own program months after being written, and encountering a statement like:

```
    170 go to 9000
or  520 gosub 13000
```

Unless intuitively obvious from the context, these statements leave the reviewer with the empty, helpless feeling of "I wonder why it goes to line 9000 here;" or "I wonder what subroutine 13000 does;" unless, of course, these lines are documented - or at least if a comment at line 9000 or 13000 explains what happens next.

It is much easier to read:

```
    170 go to 9000: rem [due to above error condition]
terminate program
or  520 gosub 13000: rem read in the next line of data
```

But it is even easier to read it without the silly word REM being in the way:

```
    170 go to 9000 terminate program
or  520 gosub 13000 read in next line of data
```

Aha! You thought those were syntax errors. Nope. Commodore BASIC allows you to put comments WITHOUT REM after most statements that end with a line number. These are the instructions that DO and DON'T allow comments after their line number parameters:

DO: GOSUB, GOTO, ON/GOSUB, ON/GOTO, RESTORE, RUN, THEN, ELSE
DON'T: DELETE, LIST, RESTORE, TRAP

The comment after ON/GOTO and ON/GOSUB must be after the very last line number.

The comment must not begin with a digit. Just like a REM, the comment must be quoted if it will contain any graphic or shifted letters:

```
    200 for j=1 to 4: gosub 1000 "Read & ignore 1st
characters": next
```

(You may be saying to yourself, "It makes sense that BASIC would ignore a comment after a GOTO, THEN #, or ELSE # since it would ignore anything after them anyway because it is going to a new line. But what about GOSUB?" Well, apparently BASIC calls the subroutine, then after RETURNing it checks to see if it is at the end of the statement (colon or end of line), and if not, it "scans" past whatever is there until it does reach the end of the GOSUB statement before continuing.)

5.8 Formatted Output [Comments 2]

Earlier we learned some concepts that allow us to control the appearance of output; that is, to format it. Among these were:

(1) the SEMICOLON (;) delimiter,
(2) PRINT FIELDS (zones) and the comma (,) delimiter,
(3) the TAB(x) function,
(4) the SPC(x) function, and
(5) the POS(x) function.

All of these are useful for formatting output to the screen, but only the SEMICOLON and the SPC(x) function could be used to format output to the printer. Furthermore, all of these have been rather rudimentary.

A much more sophisticated method of formatting output is available for both screen and files with the PRINT USING instruction for screen, and the PRINT#1f, USING instruction for files (or devices). A string parameter in the PRINT[#1f,]USING instruction tells BASIC how to print each of the items in its "print list". As an example, let's tell BASIC to print the value of the variable A, which is storing the value 57, as four characters. In order to illustrate the result, we will print an X on both sides of this four character "field":

```
    print using "X####X"; a
    X  57X
```

When PRINT USING is used, the normal rules about printing a number (space or minus sign, digits, etc. and cursor right) are NOT followed. BASIC only follows the "rules" that you give it in the FORMAT STRING, which may be a string constant or a string expression. In this example, the format string is the constant "X####X". Each # represents a digit of the number in A to be printed. The number is right justified in the area represented by the four #s.

Some characters in the format string are special characters in that they tell how to print the corresponding print list item. The # in this example is one of these special characters. Non-special characters are printed right where they occur in the format string, as is the case with the Xs above.

Another example of a special character is the decimal point. If A had contained 57.6 in the above example, it would have rounded it to print:

```
    X  58X
```

Unless a decimal point is used, there will be no decimal point printed, or any digits after the decimal point in the real number. Let's use a different format to print 57.6:

```
    print using "X####.##X"; a
    X  57.60X
```

Now it printed the decimal point and the 6 after it, but because there were 2 #s after the decimal point in the format string, it printed 2 digits after the decimal point. That's OK since we know that 57.6 and 57.60 are the same thing. The entire 7 characters ####.## represent one format field in which one item will be printed (A). A format string can contain more than one format field to print more than one item:

```
    nt$ = "answer": item = -32
    print using "The ###### is: ####.#"; nt$,item
    The answer is:  -32.0
```

This example points out several things: (1) Strings can be printed with PRINT USING, and each # special character represents one character of the string; (2) The minus sign counts as a digit; (3) Even though -32 is an integer, the format field still causes it to print with the .0 appended; (4) If more than one item is used in the PRINT USING print list, a comma is used to separate the items, not a semicolon.

If a string item is too large for its corresponding format field, the leftmost characters are printed and the others are not:

```
    print using "My name is ######"; "George Jacob
Hinkleheimer Schmidt

    George
```

If a number is too large for its corresponding format field, asterisks (*) are printed in place of the number:

```
    print using "###.##"; 1005
    ******
```

If a string item is shorter than its corresponding format field, it normally will be left justified and will have blanks to the right:

```
    print using "X###.###X"; "abcd
    Xabcd  X
```

Notice that a decimal point used in a format field that ends up being used for a string is the same as a #. A string can be right justified by using a > in the format field:

```
    print using " #>###.###"; "Answer:",1005
       Answer:  1005.000
```

There is only one format field in this format string, so after printing the word Answer right justified in the 9 character field, BASIC starts scanning the format string over to print the 1005. The space at the beginning of the format string is printed twice, once on each scan of the format string. Notice that the > is treated as a # when printing a number, just as the . is treated as a # when printing a string.

The "Commodore B Series Users Guide" (Gray Book) has more information about the PRINT USING statement and the various format field characters (pages 85 to 89), which is nicely summarized in the full tutorial on the tutorial disk.

One last feature of PRINT USING is that it normally prints a carriage return after printing all the items. If a semicolon is placed after the last item in the list, it will not return the carriage - just like PRINT and PRINT#:

```
    for k=9 to 0 step -1: print using "#"; k; : next: print
    9876543210
```

It was a wise decision to make the format field characters for strings and numerics interchangeable. This allows you to use the same format for the heading of a table, and for the numbers in the table. Let's reprogram the simple table example we did back in section 3.1 - a table of the integers -4 to 4 with their squares and cubes:

```
 50 f$ = ">######  >######  >######"
 100 print using f$,"Number:","Square:","Cube:
 110 for num = -4 to 4
 120 print using f$, num, num*num, num*num*num
 130 next
run
Number:  Square:    Cube:
     -4       16      -64
     -3        9      -27
     -2        4       -8
     -1        1       -1
      0        0        0
      1        1        1
      2        4        8
      3        9       27
      4       16       64
```

Practice using PRINT USING to learn all its features. It is consistent in that it formats output the same whether to screen, file, or device.

5.9 Program Chaining [Program Flow 8]

In very large applications, a program may be too large for memory. In still other applications, we would like to take different programs we have written and put them together - with perhaps a menu to choose which program we want. Both of these cases are candidates for use of a feature of BASIC called CHAINING, a technique where one BASIC program loads and starts another BASIC program.

This topic, too, is a little too heavy for the level of discussion in this abridged version of the tutorial.

MORE ON SUPERBASE (cont'd)

by: Bruce Faierson

This column is a continuation of an article presented in the Fall 1987 issue of the CBUG Escape. If you have not read that article, it is suggested that you read it first before continuing. If there are any suggestions for future topics in this column, please forward them to Northwest Music Center, Inc.

We shall pick up where we left off, in the Winter CBUG issue, with the calc menu option. The calc option has several important functions. The most apparent function is to perform simple calculations through complex trigonometric functions. These operations are entered with the same operators as you would in Basic. You may store the results of your calculations in variables, the current fields in your record or display them to the screen. Storing your calculation to a variable or for screen output is explained in the manual and needs no further reiteration here.

The calc option is extremely useful for modifying records. Not only can you permanently modify a record, but you can also temporarily modify the currently selected one. Any modification is temporary unless you use the store option. As soon as you select a new record the temporary modification to your current record is nullified.

One of the most useful functions for the calc command is to enter a record from the programming mode. This is attained by using the clear command to blank all the fields in the current record. Then you use the calc command to set the fields in the new record to any value. The key must be entered in the new record and then you must use the store command to save the record.

The only reason to use a database is to develop reporting features to display your data. Superbase has two ways of developing reports. The first is to use the built in reporting features. The second method is through direct programming. We shall postpone discussion of the programming method until later.

The report option prompts the user for entry of the correct information. There are a total of nine prompts that must be answered to generate a report program. Since the report generator actually writes a Superbase program it can be modified and saved to be used again. The user should buy some type of layout form to develop a display on paper before trying to set up a report format. If this is not done you will waste a lot of time, by trial and error, trying to create reports with a professional look to them.

The report option first asks for the name of the file to use in the report. Note that this is not the database but a file from the database catalog. After selecting the file you must decide how the title headings should read. The bulk of the report format will be in the report detail, totals and subtotals. Since the Superbase book and manual cover the report options in detail, we will not review what is explained so well. A few parting comments will be made.

Subtotals refer primarily to the use of sorted lists and to display changes in the specified subtotal field. If you do not use a sorted list to develop specific data it will print out information of little value for analysis. The subtotal is printed everytime the value in the selected field changes.

Another interesting feature is the use of underlining or reverse video. The commands to use are @+ for a single field or text. Use @- for turning underline or inverse on and @- again to turn it off. You may also use escape codes to set your printer for the features it is capable of. Set the escape sequence to a variable and then use the print command to send the variable contents to the printer.

A final note on reporting is to determine the quickest manner to output your report. If you are outputting all the records then use the built in report feature. If you are outputting only a small portion of the records then make an hlist and use either the report option or your own report program. The real dilemma is that the output from an hlist is very slow. To create an hlist and then output from it can be slower than just reading through all the records and outputting the selected ones. Using this principle can save you substantial time when printing your reports. One thing that can not be improved at this time is to speed up the actual print output. For some reason Superbase is extremely slow outputting to the printer.

Here is a challenge that is untested at the time of this writing. It can be observed that outputting string variables to the printer is relatively fast compared to data output. If you were to use an array to store data read from the disk and then dump this to a print buffer it might improve the output speed. Once the array has been dumped to the buffer the drive would then operate concurrently with the buffer and printer. The only limitation would be if the buffer filled up.

Another untested theory for SuperOffice users could still use the buffer. You would set up labeled blocks in Superscript for data output from Superbase. The concept is to set up your report header as a Superscript file or in Superbase variables. Then you could output your data from your list or the complete database to Superscript. The only overhead here is the automated switching between the two programs. If the theory is correct a print buffer and a fast printer should speed up operations substantially.

The last comment on the built in report generator is the lack of automatic page numbering. The only way to set up page numbering previously, was to develop your report through user programming. The page length and text length had to be set to the correct values. Finally a counter was needed to increment for the records printed. When the number of records printed, plus your header and footer, equal your text length, the page number must be incremented. Superbase the book details an easy method of putting page numbers into a simple report format.

The execute command allows you to run a Superbase program that is currently in memory. If there is not one loaded, Superbase will prompt you to enter the name of it. The execute command can also be used in programs. If this command is used with a program in memory and a new program is desired, type new first. This will clear the old program so that the new one can be loaded. Another way would be to load the program and press <7> to execute it. Type <load"program"> without the brackets. Using this method will automatically clear the previous program, load and execute it. User's choice!

The help option has multiple uses. The user can access Superbase help features if they are on the disk. A programmer could set up help screens for a user, through the memo option, to access when needed. Press <8> for help and then enter the name of the help screen needed. To change menu you only have to press return. The second menu in Superbase is oriented toward file structure, file and record modification and disk maintenance. Both menu 1 and menu 2 have access to the help command by pressing <8>.

The first option is the file option. The file option displays up to 15 files created under the chosen database name. These files do not have to be related but in practice should be. Any file that is open will be highlighted in reverse. Only three files can be open at any time. The first one opened will be the first closed if more are accessed. It is of essence to remember to open all necessary files again if you are linking files. If one of the linked files was the first open and a new file over the three limit is opened, then you would close the first file inadvertently. Another tip is to keep as many needed files open at one time. This will keep the load time to a minimum as file change time would be kept to a minimum. Check the Superbase manual for maximum system specifications for files. It details how much space is available for files in the appendix.

The format option allows the user to create custom displays for record entry. Up to four screens can be created per file and be selected by using the <screen 0-3> command. Inverse characters can be used to accentuate the display and you may select a character to draw borders with the cursor keys. This feature was extremely advanced when released as most other databases made you program your screen displays. Anyone who has done this knows how time consuming it is.

Another interesting concept for the file screen displays is discussed in Superbase the Book. To create a custom entry screen with only a few field entries normally requires programming. If you only need one screen for your entry screen it is possible to use the other three screens for custom screens or menus. If used in this manner the display is incredibly fast. It seems that the screen displays created through the format option are not interpreted as are the basic display statements. If you are programming large displays with a lot of descriptive text you will appreciate this feature. Your computer does not look like a slow terminal when scrolling the screen.

There are two minor drawbacks in the use of this option. Firstly, you must have an unused field in the screen format before you can save the screen. This means that the field will reveal itself anytime you call that screen display. To eliminate this unpleasant side effect, you can display over the field. First, pick the location of the field and then use the display command to cover it over. This will eliminate the field until a new screen is selected. If you display a prompt over the field it is very effective and should be done right after the screen is selected.

The second dilemma is if you need three files open you have to be careful how much text is used in your displays. If you use too much descriptive text it may exceed maximum specifications and cause the closing of a file that you need. This can cause extreme sluggishishness in your system.

The batch option operates on all tne records or a list of records previously defined. I am not sure whether this option is any faster than direct programming. It does not seem to be much faster than writing a simple algorithm to select each record, perform a test and calculate a new value for a field. I think the value of this option will depend on the size of the list that is used. If it is a straight forward update of all the records in a file the batch may require the least programming.

The sort option is extremely useful but can be tediously slow. It is not appropriate if quick reporting is on your mind. The basic premise of a sort is to create a list of the key fields in the sort order. It is not a whole new file with all of the individual fields from each record. This concept saves disk space and it should really operate much faster than it does.

To illustrate a simple example of its' usefulness, let us look at an accounts receivable application. If you need list of items more than 30 days past due and have them printed in high to low amounts you should develop a strategy. You should determine by past experience whether the number of records involved in the report is a substanstial part of the database. The reason for this is as follows. For this report you have to find a list of all items 30 days past due, sort on amount and select from the sort list to print your report. Sounds simple but this can be very slow for a large list of items. It can be so slow that you may want to think about just printing a list of past due items in any order. It is all relative to the time factor.

If time is not important or the list of items is small, the best way to achieve this report is stated above. The find option will give you your basic list of items and the sort option will restructure your sort key list from high to low balances. Then you must report on all items from your sort list. You will have to work with this option to see how you can develop the fastest and most efficient method of using it.

The next option is the program editor. We will not discuss programming here but will begin to examine this extensive subject in the next installment of this series. The program editor allows you to develop your own automated applications. Although you can do everything from the command line, it has been said that a program will be developed anytime a programmer has to do the same thing twice. Don't waste your time typing in the same instructions for the same reports every time you need them.

The program editor allows you to enter 179 characters per line and allows you to scroll the screen forward and reverse. This is a nice feature. The capability of reviewing a previous line in CBM Basic, that is not displayed, is not available. The only flaw to this editor is that you may not edit anything on the second line without cursoring through the first line to get to it. The programming option is one of the best reasons to use Superbase. The applications that can be developed are only limited by your imagination and ingenuity. To exit this option type <esc> <q> without the brackets. Make sure you save the file by typing <save"file"> without the brackets.

Contrary to the C-64 copied manual, you have 8k worth of program and 8k of strings and arrays available in the B-128 version. A word to the wise is to check your available programming space by typing < display fre(1) > at the command line or set a function key to do this automatically. Do this frequently anytime you are nearing the limitations of the programming area. If you do not do this and save the file you will be not be able to load it in Superbase again! Expect to spend your time using Super Disk Doc to reduce your file size. One last suggestion is to print your program anytime substantial modification are made. Type <print:list:display> without the brackets.

The next option, maintain, is the disk and file utility section of Superbase. This option is extremely important for bringing in new data from another type of system, sending data to a sequential file for archiving and executing CBM disk commands. All of the important options can be executed under program control. It is very useful to be able to automate your disk commands and file exports for backup purposes. Use the functions with care as the scratch command will not verify your intent under program control. The maintain options and their intended functions are as follows.

1.) status - This option will allow you to print or display the field names, their lengths and the types of fields in the current file.

2.) catalog - This option prints or displays the list of files in the current database.

3.) import - The import command is used to bring data from another database into Superbase or data from a totally different system into Superbase. An example is to bring in data from Dbase through a modem or null modem. The terminal program can convert the true ascii to CBM ascii upon request. The data must be in CBM ascii format and with carriage returns after each item. The file format must be the same as the one you are importing. If you have a delimiter other than a return between records you must specify that delimiter in your import command. A delimiter is a character that is not normally used in a data field. This character tells the importing program that it has reached the end of the current field and may start with the next.

4.) The export command is used to extract data from a database to be used for backup, to import data into a new Superbase database or to send the complete file to another computer with a file of the same format. This function selects all the records and fields. It may not selectively pick fields. The output option is for selective disk file output. This option can also add a delimiter to replace the carriage return• that is by default after each field.

A practical example of using this feature is if you were working on two different computers, one at work and one at home. You could have two totally different database programs such as Superbase and Dbase but with the same file structure. You could export out the data from Dbase and telecommunicate the file through a modem to your system at home. Then you could import the file into Superbase to work on at home. When your work at home is completed the process can be reversed to replace your file at work.

5.) directory - gives you the directory on the current logged unit and both drives if available.

6.) backup - makes an archive copy of your disk. The single copy option on an SFD is only functional under Superbase 2.

7.) new disk - creates a formatted disk from either a previously formatted disk or a new disk. If you have specified a database on drive 1 make sure that the new option does not default to that drive. I prefer to new disks out of Basic for safeties sake. One interesting possibility of this option is to execute it under program control as a protection device.. This is assuming you have archive copies elsewhere. The concept is to have a password protection scheme and if the user can not get in within a certain number of times to have the data disk erased. This will give you control over sensitive data. USE THIS IDEA WITH CARE!

8.) The <other> option gives you the ability to scratch, copy and rename files in shorthand compared to Basic. There is also the capability to validate the disk and do directory pattern matching. Remember, under program control it will scratch files without your getting a second chance. Watch any pattern matching on this option.

The memo option can be used to create help screens, write unformatted letters and messages and to create menu screens to call from a Superbase program. The memo can include up to four screens and may be edited page to page. For further editing instructions check pages R-79 and R-80 in your manual.

The last option is the help option. From this selection you may get help on all of the main Superbase functions. You may also create help screens with the < h > prefix that a user may access when needed. This is very handy when there are multiple options that a user may request and to explain them in a help file. Help screens may be printed by pressing <shift><run/stop> at the same time.

Well, this does it for this issue. We hope that these articles are giving the reader some insight into the personal tribulations that the writer has endured in learning Superbase. It is hoped that some of the mysteries for the non-user have been cleared up and maybe the experienced user can get a fresh perspective on some of its' functions. Please note that these articles are not meant to rehash what is already in the manuals but to embellish what has already been written.

On a final note, NWM Inc. has devised a very fast and efficient way to use a dual index scheme in Superbase. As many of you know, Superbase allows only one key per file. There is no way outside of rewritting the program to have a second key field. We have found a way to achieve this with only slight inconvenience. This is not the slow and cumbersome method as offered by the ICPUG group in the last issue. Accessing the second key on a 9090 only takes two seconds. We will cover this in the programming articles.

◆◆◆◆◆◆◆◆◆

B-128 SERIAL BUS

by: Dennis Jarvis

Now that we have seen the release of the B-128 serial bus, we thought that it would be a good idea to give you more information of the capabilities of what this HARDWARE/SOFTWARE duo can do for you and some suggestions of possible uses.

With the B-128 serial bus you will have at your finger tips a full blown serial bus just as it sits on the other COMMODORE computers from the VIC 20 to the C-128. With this new addition to your computer you will be able to transfer data, and programs to/from other computers much easier than it was before. Using your existing software you will be able to copy TEXT files from a C-64 for example to your B-128 using a serial disk drive such as the 1541 or 1571 to your IEEE disk drives such as the 4040 or 8250 using your existing programs such as COPY ALL, or UNICOPY.

Along with TEXT files you can transfer over your SUPER BASE data from one of the other COMMODORE computers such as the C-64, to the B-128.

One of the main advantages with the B-128 SERIAL BUS is it fully gives you access to the FAST BUS capabilities of the new 1571 disk drive which gives you quicker loading times on program files. For example if a 100 block program takes you 14 seconds to load off your 8050 disk drive, the same program will load up in 6 seconds with the 1571 disk drive, and 4 seconds with the 1581 disk drive. Thats over 50% faster that the IEEE 8050 disk drive!

For those of you which do have access to the new 1571, or 1581 disk drives and are writting programs to use the new BURST MODE COMMAND SET (BMCS) then you MUST take the time to look over our massive FAST BUS documentation which will greatly assisst you in using the BMCS commands.

We have loaded up most of the BASIC programs off of the 1540, 1541,and the 1571 test demo disks and found them to run error free with NO changes required to run on your B-128 computer.

Jim and I have made every attempt at making your B-128 SERIAL BUS as simple to use as possible, with just a "FLIP" of a switch you can access either your SERIAL BUS devices or your IEEE devices.

Well that's it for now until next time...

A NEW DESIGN CO-PROCESSOR

By Gary Anderson

IMPORTANT NOTICE! If after reading the following article you feel this is a worthwhile addition to the continuing effort of enhancing and improving the low profile B-128's capabilities and features then please write me a short note to help me decide if I should layout a circuit board and offer a finished product for sale for about $339.

A LONG TIME: For the past year I have been working on the design of a new V-20 CPU co-processor with some significant improvements and modifications over Commodore's original 8088 CPU design. It has taken quite a while to analyze the old schematic, learn the 8088 and its support chips, modify some of the logic, develop a new schematic, purchase the new parts, build a handwired prototype and get it debugged. The V-20 processor chip is software compatible and is a direct plug in replacement for the 8088 chip. This new design V-20 co-processor is useable with an adapted and modified MS-DOS and CPM-86 operating system. A schematic and software listing of the boot rom of the old 8088 co-processor design can be obtained from CBUG, Archive #001.

OUT WITH THE OLD: Let me describe some characteristics in the old 8088 design that in my opinion required some attention.

First, Commodore's 8088 co-processor design fits nicely in a high profile but does not fit very easily in a low profile. Since the majority of the membership has the low profile, along with myself, the board layout would have to be changed. Unsoldering the parts from the old board and taking a "picture" of it to abtain negatives of the layout just does not cut it.

Second, the old 8088 co-processor design does not address the 256K by 1 DRAM memory chips which are presently used by the 1 megabyte memory implementations, so another IC would have to be added to generate the added EXTMA8 (external memory address #8) address line. Four rows of 64K for a total of 256K on the main board would otherwise be the max system and then it would not be compatible with any upgraded 1 meg memory machines that have greater than 256K of memory.

Thirdly, the +5V current consumption on the original 8088 co-processor board comes in at exactly 1 AMP, OUCH! The B's switching power supply already runs hot and does not need to run any hotter. This would definately take away from any future enhancement ideas.

Lastly, the clock frequency on the 8088 co-processor is 5.0 MHZ. Granted, that is faster than the IBM PC at 4.77 MHZ, however it takes 4 clock cycles to read a memory location reducing actual single byte through put to 1.25 MHZ. The B-128 itself has a single byte through put of 2.0 MHZ.

IN WITH THE NEW: I now have a hand wired prototype V20 CPU co-processor working in my low profile B-128 that makes improvements on the original 8088 CPU co-processor's above mentioned characteristics.

First, the dimentions of the board were chosen to allow for the installation inside a stock B-128 or on the B-1024 1 Megabyte Memory Expansion Board with the top cover completely down. To implement a maximum number of features, added circuit boards must stack neatly under the hood and not use ribbon cable to interconnect them.

Second, I have added another IC to generate EXTMA8 which makes it compatible with the B-1024 1 Megabyte Memory Expansion Board, I can not speak for the other 1 meg upgrades as I did not design them. This added IC causes no problems when used with a stock B-128's 64K DRAM chips.

Thirdly, to reduce the +5V current consumption, where possible, NMOS parts were replaced with CMOS parts along with LS and S series TTL being replaced with ALS and AS series TTL. I went with NEC's V20 CMOS processor which is 8088 compatible, CMOS 82C84A clock generator, 82C88 bus controller, 82C59 interrupt controller, and a 27C64-20 UVPROM. The drop in +5V current was dramatic, the Commodore board draws about 1000 milliamps and my new design draws only 385 ma.

Lastly, in an attempt to increase the speed of execution a redesign of some of the timing circuitry was done. Row and column address strobe logic for the DRAM was reworked to allow for a speed up in clock frequency without having to change to faster DRAM or add wait states. Changing to faster DRAM would be messy considering the 64K memory chips on the main board are soldered in. So far a clock speed increase from 5 MHZ to 6.2 MHZ has been realized with the prototype. Granted this might not sound that great but it is noticable on the screen.

As stated earlier if there is enough interest I will layout a circuit board so please let me know your comments, thoughts, and ideas at the above address.

Gary L. Anderson
2560 Glass Road N.E.
Cedar Rapids, Iowa 52402

◆◆◆◆◆◆◆◆

HARDWARE RESET and
8050 OPERATION OF/CONVERSION TO THE 8250 DISK DRIVE

By: R. D. Connely

Original concept and article printed in the Joplin Commodore Computer Users' Group Newsletter, V2, N2, February 1985.

I, like many of you, purchased my B128 system from Protecto at what seemed at the time a give away price. Unlike most of you, history has shown, I had long since purchased an 8250 2-meg drive years before for my SP9000 system. The files I was keeping on the SP were becoming cumbersome and I was looking for some elbow room. The B128 + 128K expansion memory provided that room, however, that 8050 drive was a real nuisance. It couldn't read the data disks I had already produced, yet I felt having two of the big drives on line, or at least having a backup, would be beneficial.

My 8250 uses Micropolis 1006-IV double-headed 100 TPI drives, but the 8050 used MPI 101 drives. The disk controller chip at IC UK 3 is different depending on the type of drives being used. It is CBM #901885-04 for the Micropolis and Panasonic (SFD 1001) drives and is #901869-01 for the MPI drives. Not wishing to replace this chip, I began by purchasing two MPI 102 (100 TPI, double-headed) disk drives from a West Coast parts house and converted the 8050 to an 8250. A minor logic board change, rewiring the analog harness, and head realignment has required. The head alignment was the most tedious, Physical Exam 8250 hadn't be invented yet. (This information was printed in the Joplin Commodore Computer Users' Group Newsletter and noted in Issue 22 of the Midnight Software Gazette & Paper.) This was not the end of my problems, though. I had had compatibility problems with the 8250 from day two. Jim Butterfield's Copy All program wouldn't copy from 8050 disks and so I had to modify it to ignore the 8050 disk's "66, illegal track and sector" error message. Later came Knight's 8050 Backup program which refused to work in an 8250 drive, period. Curses on everyone who wrote anything for the 8050 disk drive that didn't consider the 8250 drive, also! Shame, shame, shame! This seed of incompatibility eventually lead me away from the B system and its really wonderful drives. <<Simple answer, insert a SPST throw switch in series with Jumper E3 as described below -- located immediately above UL2. Configuration will be selected at power up.>>

The minor logic board change I mentioned above is the heart of this article. Remove all connectors to the mother board (the one in the lid) and carefully remove the board. Make your wiring connections on the back of the board. A trace leading away from pin 18 of IC UK3 (disk controller) about 1/4 inch long terminating in a thru-board hole must be cut. Directly above IC UL2 about 1/4 inch is a thru-board hole and 1/2 inch directly above it is another. If these holes are jumpered, leave it. If not, install a jumper. Why? I don't know. <<According to CBM schematics, the jumper (jumper #E3) determines 8050/8250 election at time of powerup. Installed for 8250, open for 8050. Suggest using switch. This switch is ALL that is required to run an 8250

in 8050 mode. Note: The current Schematic shows Pin 18 as no connection (probable drafting error) and is labeled "not reset" i.e. a low state forces reset. Jumper E3 is shown between Pin 21 of UK3 and Pin 1 of P2, which inturn goes to a drive select pin on the analog board.>>_ I can tell you the mother board in my 8250 drive had it and the mother board in the 8050 didn't. The 8050 seems to work ok in 8250 mode without it, but it seems safer to me to install it. Now solder a 2 foot long wire to pin 18 of IC UK3. When this line is tied to the return (0 vdc) side of +5v, the mother board is in 8250 (double-sided) mode. When this line is open, it is in 8050 mode. However, like the address select lines at UE1, this jumper is only read at power-on or at reset. Now, it is obvious to me that we want to change modes at will, without resetting the B or a printer, etc. So we must also include a reset that will work only on the disk drive. On the PET we always grounded the reset line at the 555 timer. Yep, we have a 555 (IC UM2) here, too. On my boards there is a 10K ohm resistor leading away from the reset line, pin 2. Solder another 2 foot long wire to pin 2 of IC UM2. Find any convenient thru-board hole in the 5vdc return trace and solder another wire which will dress out in length to the other two. I drilled two small holes in the mother board 1/2 inch apart near UK2 where it didn't damage any traces on either side of the board and wire tied the now-bundled cable to it as a strain relief. I mounted a single-pole, double-throw, momentary-one-way, center-off micro-toggle switch in the cabinet where ever it was convenient (between the drive faces on the lower chassis on one unit and just out the side of the top on the other). The 5vdc return wire is wired to the center of the switch; the reset line is wired to the bottom so that it is active when the switch is flipped up to the momentary position. The wire from pin 18 of IC UK3 is wired to the top terminal so that it is active when the switch is flipped into the locking down position. A quick flip up on the switch causes an internal reset of the 8050/8250 drive. Note that both circuits cannot be shorted at the same time and neither are shorted when the switch is in its center position. By flipping the switch up you initiate a reset and pushing the switch down quickly the drive will read the switch as indicating an 8250 mode. Reseting and allowing the switch to rest centered will be read as indicating an 8050 mode. This will allow you to correctly boot and load from 8050 disks, yet go to an 8250 mode for data disks, etc.

The same fix may be installed on the SFD 1001. Although I didn't need this modification, myself, I was curious. The 555 timer is at location 11J. You still ground pin 2 for a reset. The 901885 disk controller chip is on a daughter board along side of a 2716 EPROM and a 7414 TTL gate. Do NOT make connections to the daughter board. Remove the mother board, cut the short trace on the bottom leading away from pin 18 of the socket into which the daughter board plugs. Find any convenient spot to attach the 5vdc return wire and bundle the three wires together. Strain relieve the cable somewhere and wire the same kind of switch in the same manner as described for the 8050. By the way, the unit address selectors are at IC 4J, in case you want to cut some and/or switch some of these, too.

I don't recommend these modifications to the faint of heart or to the inexperienced. Cutting traces and soldering to MOS circuit boards requires care. I recommend static wrist strap, grounded 35 - 40W soldering pencil, Kester 60/40 .050 solder and removal of the chips in the sockets you will be soldering to. I also used a hot glue gun to dribble glue here-and-there to retain and insulate the added wiring.

I highly recommend Physical Exam 8250 to test out the speed control and tracking of your drives. I had to add the 5014A diodes to the MPI controller card just to use it winter and summer. It still isn't perfect, but is tolerable. Oddly enough, the Micropolis drives came with the 5014As but has been 100% always. I haven't noticed any speed problems with the Panasonics, either.

Goodluck! -rdc.

CABS PAYROLL OPERATIONAL INSTRUCTIONS

By: Lowell Breunig

To run the CABS/INFO DESIGNS PAYROLL program successfully you must understand the use and function of the date fields that appear in 01 P/R Information file. Today's date, Period start, and Period end. See page II-3 of the Payroll manual. The manual is not very clear, but it dose help some.

(Today's Date) date field:
Has two functions. It appears as the report date on all printed reports. More critically it is the date that appears in the date field at the bottom of 02 Maintain/Display employees, AFTER 09 Compute Gross Pay and 10 Compute std. Deductions has been run.

THE PROBLEM:
Today's Date acts as a key to summing all data in 12 Payroll Journal and causes what appears to be summing errors in the Company Totals option of the report. During the summing procedure the program looks at all the employee records and sums all the records that have the correct date (Today's date). So far so good, but what happens to the data paid the the guy you canned last pay period and didn't require entering data to this pay period?

Because the record was not updated it will not contain the correct date. Therefore is not summed and the Quarterly and Yearly totals are understated. To solve this problem it is possible to enter 02 Maintain/Display Employees and change the date field to the date of the report (Today's date) on all employee records. Ok so now the Quarterly and Yearly totals are correct but the current amount is now over-stated. Why? because the current fields contain the data from the last check of the guy you canned last pay period! So we'll be real smart and re-enter 02 Maintain/Display Employees and zero all the current fields on his record and now everything will sum correctly; right - WRONG! Why? You didn't zero 16, through 18 of the HEADER INFORMATION screen, the first one that comes up when sub/menu 2 maintain/display is selected from Maintain/Display Employees menu.

SIMPLE SOLUTION:
All of the above can be avoided if you enter a zero or a simple carriage return during the next pay period. That action zeros all current fields. To get accurate Quarterly and Yearly amounts on the Company Totals report, you still must change the date fields because sending a zero to all X-Employee's will leave 99/99/99 in the date field. If you have a lot of employees this is impractical. I hope this problem has been corrected with the up-grade and if not it should be.

A NOTE OF HAPPINESS:
The Quarterly Tax Summary is not effected by the above and uses the correct Gross Pay amount and can be used on quarterly reports. Also remember that the above only effects the Company Totals report, all the data on the rest of the Journal report is correct.

(Period start) date field:
Period start, dose control the monthly, quarterly and yearly cycle processing. The problem is understanding the "period" referred to. Assume we use a monthly accounting report cycle and pay bi-monthly. Then the period start date would be set at the beginning of each month, NOT THE BEGINNING OF THE PAY PERIOD. You MUST also change the PAYROLL NO. at the start of each PAYROLL PERIOD. If you mess up and forget either one go back to your last correct back-up disk and re-enter all data.

To restate in another way:

1st to 15th pay period, set PERIOD START to 01/mm/yy and PAYROLL NO. to 1

16th to end-of-month , set PAYROLL NO. to 2 and PERIOD

START remains the same.

This would be correct for a Monthly Accounting cycle or period and bi-monthly pay cycle or period.

(Period end) date field:
This date simply lets you print reports and checks with two separate dates. Assume you pull your payroll on the 15th of the month and actually pay five days latter, then the Period start date would be 05/15/88 and the end date would be 05/20/88. If you printed your checks on the 15th they would be dated mm/20/yy and reports would be dated mm/15/yy.

(Payroll no.)
It is critical to change this number for each dispersment of pay. If you pay twice a month on then the first pay period (1st to 15th) would be 1, and the (16 to 30) would be 2. If you pay weekly then each pay period would be 1, 2, 3, 4, and 5 if required. You must not change the period start date until the monthly pay cycle is complete.

A note of caution. The above procedure applies only to Non Up-graded Disks. I have purchased one but have not played with it yet and I understand the there were changes made to this and the date areas. Also as I understand it the 5th week period didn't work. As I only require two I have not tried this option.

EMPLOYEE TYPE:
You must place a 99 (inactive) number in line 09 of Sub-Menu 2 Maintain/Display or Absentee and Insurance reports on disk-B will contain data from these non active employees.

BUG M
If you want two copies of the MISC. PAY/DEDUCTIONS, you must drop out of the program and reload it or the second total will be added to the first.

I have been using CABS/INFO DESIGNS Payroll successfully for a two and half years. I would suggest you follow the E. page IV-7 of the payroll manual as an operational procedure and I further suggest that you make any corrections such as a pay increase to employee files as your normal, no. 2 function. Following the list with the one addition and understanding the above date fields and print-out problems will allow you to use this program successfully. I also recommend that you run a Journal Report with current, quarterly and yearly totals during each pay period as it allows you to rebuild from that point or re-enter anything you wipe out while mucking around in 02 Maintain/Display employees files.

If anyone needs help with it I will be glad to help. Call 1-503-630-6591, best time is 9:30 to 10:00 AM or 9:00 to 12:00 PM, (Pacific time). I have just purchased the upgrade and hope the above problems have been corrected, if not it should be relatively easy to correct once the problem is understood. I would have written this before, if I had known others were so troubled, sorry.

Lowell Breunig
950 N.E. Hillway
Estacada, Oregon 97023
(503) 630-6591
9:30 AM Pacific time is best

RESTAURANT ACCOUNTING

TSUM and MLOG

By: Lowell Breunig

TSUM and MLOG are companion programs written for use in a Restaurant, however they can be used by other business or private use. TSUM is an internal invoice, Sales and theft control program. It's purpose is to supply operational control data on daily basis and retain that data for future use as needed. Data input from source documents, is fast, with easy corrections performed during input.

TSUM, it is be impossible to describe TSUM in a paragraph. Two of its best features are the ease and speed of input with instant price verification. The simplicity of number series tracking, such as invoices, will blow your socks off. Anyone who needs to control Guest-Checks, Invoices, Purchase Order forms or any other need to track items in series should see this routine. It's simplicity in itself, and there isn't a need to read a long list of numbers or have reams of paper gathering dust.

MLOG and TSUM are companion programs, however MLOG can be used as a stand alone program. The purpose of MLOG is to set up a list of items within 14 categories with a unique code number for each item. It can be used as an inventory list, phone book and/or address book. It is easy to run, numbering is automatic and allows additions or corrections to existing lists or just brows through your records, forwards or backwards with the touch of a key.

If you are learning to program these programs have a number of routines that are easy to pull out as a block and play with. All the major variables are listed in rem statements at the top of the program and all routines separated by rem statements.

None of the commercially available Restaurant programs, I have seen, allow changes to the pricing of a menu item if you desire to modify it for a customer. Making the customer happy is our primary function and a rigid system dose not work. Commercial systems are unfortunately big on accounting, but don't forget that Balance Sheets and Income Statements, though useful, are primarily for use by others outside your business. It used to evaluate it for loans or purchase. I was more interested in operational information to control cost, enhance sales and control theft, etc. Commercially available programs costing $10,000 plus for software and requiring new hardware, and then didn't give me what I needed most, was a little to much. TSUM and MLOG were developed supply operational information. We have been using them since 1986 and I am pleased enough to share them with you as long as your not an immediate competitor. Because this program is run by the Accounting Dept. or the Owner in smaller Companies, there is little chance of the Wait-Staff being able to have unaccounted for guest-checks and/or get into undercharging in order to earn bigger tips. For those of you not aware of the Restaurant business, these are serious problems. It is a big enough problem that many Restaurants have failed by not controlling the problem.

This is a little off the subject but will demonstrate the need for this or similar programs and computers in general. A restaurant with 200 customers a day, (that is small for the restaurant industry) ordering an average of 4 items each, will purchase 1000 separate items a day. If tracking each item takes 1 minute to process it requires 17 hrs a day. TSUM cuts this time to 4 seconds average or about one hour a day. Even worse are losses due to theft, they can send you to the poor-house in a hurry, consider this. One waiter a day who thinks they deserve $20.00 more and pockets that Guest Check has the following impact on finances. The $20 will cost the business and additional 60% in inventory replacement, direct labor put into the product etc.. The $20 is now a cost of $32. Many restaurants operate 365 days a year, therefore the potential loss is $11,680. If you have five of these soles total potential loss of operational funds will be $58,400 a year. Still think you can run it by the seat of your pants. When we first opened our restaurant we lost $19,200 over a five month period from theft alone. Interestingly your employees, customers and business do better and are happier, if the staff is not worried about getting caught and they earn their money making the customer happy. We have been in business for 9 years this month.

<<Note: The text files on the library disk are in Superscript III. In Superscript II they can be read on the screen in editing mode without modification. To print to paper will require minor modifications of the * commands. This is easily done using the F1 and F11 keys.>>

THE CBUG LIBRARY

We certainly have something for everyone this issue. Tony Goceliak has given us a highly advanced editor, a disk doctor, and much more. Mathew Goldstein answered the programmers SOS with his Syntax Checker -- and a bunch more.

For the business person, we have an extensive but easy to use Restaurant Accounting program from Lowell Breunig. Remember, just because it says restaurant doesn't restrict its use. Moreover, specific routines can be adapted to numerous unrelated applications.

Entertainment and Useful Programs. Fred Peterson has outdone himself with his two disk production, Gold Coast Gander. Everything from games to utilities, construction (building) estimation programs to financial computations.

How does the B-128's subsystems relate to each other? If you are fixing a B-128 or writing an exotic application, then look to John Plosila's extensive technical notes for answers.

As always, Lt. Col. wright has provided more interesting materials for members using the 8088 co-processor.

Do you enjoy the library disks? If so, it would be nice to send a note to the author. The contributors spend literally hundreds upon hundreds of hours preparing the library disks for the sole gratification of knowing some few members in the group may have use of them or enjoy them. A word of thanks isn't so much to ask, is it?

The CBUG LIBRARY is now well over 100 disk titles. To avoid confusion, we have used Superbase to generate lists sorted by CBUG numbers and by stock numbers. These are now included on the order forms.

AN IMPORTANT TIP

The recent release of The Bible on disk has brought to fore the fact that you can not correct and resave a Superscript file to disk if the disk is full -- which The Bible disks are. The reason is that when replacing a disk file in Superscript, the corrected file is first saved, then the old file is deleted. The corrected file leaves its directory title in the original position however. If you need to replace a SS file on a full disk, you must first scratch the original file. Then when you save what you have in memory, it will go back in the same position on the directory.

A FEW NOTES: We have experienced a small number of defectively duplicated library disks being shipped. This problem was finally traced to a uniquely defective 8050. It had the unbelievable habit of occasionally modifying a random bit or two on the master disk! That is supposed to be impossible, but it happened anyway. With luck we have caught all the remaining bad copies and masters. Thanks for bearing with us -- and letting us know when there is a problem.

When sending disks to CBUG be sure and mark them well. ALWAYS write the current date on the title label per chance there may be a later update, etc. Always include atleast your name, zip code, and phone number on a label on the disk. It is very important that all of the above be included as identification on the disk itself.

GOCELIAK's GOLD MINE #0788 CBUG #88 NEW RELEASE 2802 #13090

As usual, Mr. Goceliak offers more and more important discoveries and insights. The Mr. Ed program is reported to be the most advanced disk doctor type progam yet written for any Commodore machine! While no one except a dealer would likely have 27 disk drives, Tony shows us how to boot up as many as we do have each with its own device number. AND, true to form, more truly valuable test programs. Many of the instruction files are printed this issue

```
34    "mr. ed"            advanced b series disk editor - any ieee drive
48    "mr. ed.ins"        documentation file for above
0     "----------------"
2     "&reback0 to 1"     four files to produce backups much more quickly
2     "&reback 0to1+"     from drive 0 to drive 1
2     "&reback1 to 0"     or from drive 1 to drive 0
2     "&reback 1to0+"     optimized for minor or major disk changes

2     "&certify bac+"     a way to tell if a 'backup' disk is up to date
36    "reback files.ins"  documentation for above disk programs
0     "----------------"
1     "&drv adr #4"       a series of disk programs to alter the unit #
1     "&drv adr #5"       with instructions on how to configure an
1     "&drv adr #6"       arbitrary number of drives all powering up
1     "&drv adr #7"       as unit number 8 with a single command from
1     "&drv adr #8"       you.
1     "&drv adr #9"             1    "&drv adr #20"
1     "&drv adr #10"            1    "&drv adr #21"
1     "&drv adr #11"            1    "&drv adr #22"
1     "&drv adr #12"            1    "&drv adr #23"
1     "&drv adr #13"            1    "&drv adr #24"
1     "&drv adr #14"            1    "&drv adr #25"
1     "&drv adr #15"            1    "&drv adr #26"
1     "&drv adr #16"            1    "&drv adr #27"
1     "&drv adr #17"            1    "&drv adr #28"
1     "&drv adr #18"            1    "&drv adr #29"
1     "&drv adr #19"            1    "&drv adr #30"
13    "drive unit adrs'"  documentation for above.
0     "----------------"
7     "90xx test pgm"     a way to detect whether it's time to re-header
                          a hard drive by sensing read degradation.
12    "90xx testing.ins"  documentation for above.
8     "80xx test pgm"     a similar program adapted to floppy drives
5     "80xx testing.ins"  documentation
0     "----------------"
14    "runmode auto-vfy"  a way to ensure that the basic program you
                          just loaded was CORRECTLY loaded without fuss.
5     "test read 40vfy"   demo program of above technique
0     "----------------"
8     "9090prot cpm.gen"  If you have a hard disk and run cp/m you need it
10    "9090prot cpm.ins"  documentation file for above
0     "----------------"
1     "more articles"
27    "duplicate fnames"  how to use duplicate filenames and why.
4     "dos sort number2"  additional use for the 'uncommon sort'
9     "no name files!"    how to create and use files with no name at all.
38    "my new pup"        programming pups not pets
0     "----------------"
1     ""                  the no - name demo file to dload and run
0     "----------------"
11    "blurb"             this file
1722 blocks free.
```

SYNTAX CHECKER - PLUS CBUG #89 NEW RELEASE 2812 #13118

By: Mathew Goldstein

This disk, CBUG #89 also includes the entirety of CBUG #90 as well without additional charge.
The SYNTAX Checker program is explained in great detail in an article,this issue (Vol 11, Spring 1988)

Some additional comments:

' Mastermenu v2.3' now includes commands to alphabetize the menu and to compare the contents of the menu with the disk directory. It could be improved, if anyone is interested and up to the challange, by using customized m/l subroutines to speed up directory reading and directory alphabetizing.

I thought it a good idea to have a second, small menu program that is compatible with ' Mastermenu' which loads files only. The first file on your disks should now be ' Midgetmenu2.3/c' followed by ' Mastermenu' and '+print screen'. To initially set up your disk run ' Mastermenu' (see the instruction file). From then on you can start with ' Midgetmenu'.

'compressor.mg' will add ':else' tokens to terminating 'if ... then ...' statements for more thorough compacting. It could be further improved by dropping unnecessary repetition of 'print', 'print#', and 'data' statements on the same line that occur as a result of combining consecutive lines. Also unnecessary semicolons in print statements and unnecessary paranthesis in string or numeric expressions could be removed. For best results when using the 'compressor.mg.gb' program, first use 'uncompact' (CBUG #4 and #32) and then compress the uncompacted program. Try compressing "%checkbook v3.3" and other long BASIC programs to reduce loading time.

If you encounter two alternative ways of programming a routine use the stopwatch feature of 'alarm clock.mg' to find which routine is most efficient. Or keep the alarm clock running to give your eyes and back a periodic change of pace.

'basic compare.mg' accurately displays the actual differences, if any, in two programs. 'f-key saver.mg' uses memory addresses to efficiently save the f-keys without affecting the screen display. 'optimal value' uses linear programming techniques and does limited what-if analysis. 'underflow' detects an underflow condition for computer multiplication and division which can be very useful. 'dekker.sample' runs on the 8432/e emulator and demonstrates an implimentation of dekker's algorithm for mutual exclusion when running two concurrent processes.

You will know that you BASIC programs are syntactically correct when you own 'BASIC 4.0 syntax'. You will be satisfied with this fine program, which can be purchased through CBUG for the bargain price of $26. 'BASIC 4.0 syntax' will also optionally search for unutilized lines. Do not rely on my previously published public domain program 'unutilized BASIC' which is flawed.

If you own a compiler it is a good idea to run 'BASIC 4.0 syntax' on programs which you are going to compile, before you compile. You will save time by compiling to completion the first time and avoiding unnecessary re-compiliation. The Petspeed compiler is very good despite it's assorted flaws. A better compiler would allow for optional run time array bounds checking, facilitate interfacing compiled subroutines to BASIC programs, provide better error detection and notification during compilation and at runtime, provide speed/size optimization options, and be capable of compiling all BASIC 4.0 code without exception. What we really need, though, is a compiler for another programming language such as C++ or LISP or Modula-2 or PROLOG. The 'B' is a solid machine, but it's full potential cannot be realized without better programming language support.

```
1 "BASIC 4.0 syntax" 1d 2c     61   "basic comp.mg.gb" prg   4    "dekker2.sample"   prg   14   "Comments.mg"      seq
12   " Midgetmenu2.3/c" prg    11   "compare dir's.mg" prg   4    "dekker1.sample"   prg   46   "BASIC syntax.ins" seq
1    "+print screen"    prg    1    "f-key saver.mg"   prg   47   "tableofss2graphi" seq   33   "checkbook3.3.ins" seq
1    "B syntax loader"  prg    3    "day from date.mg" prg   127  "BASICsyntaxbatch" prg   37   "Master2.3.ins"    seq
129  "BASIC 4.0 syntax" prg    32   "optimal value.mg" prg   7    "Masterdirectory"  rel   1    "Bsyntbatchloader" prg
6    "alarm clock.mg"   prg    6    "checkbook v3.3"   prg   68   " Mastermenu v2.3" prg   1144 blocks free.
62   "compressor.mg.gb" prg    185  "%checkbook v3.3"  prg   1    "From.mg"          seq
1    "+alrm clck.f0400" prg    2    "underflow"        prg   6    "Annotation.mg"    seq
```

GOLDSTEIN's ASSORTMENT CBUG #90 NEW RELEASE 2811 #13122

By: Mathew Goldstein

See the descriptive text in the listing above for CBUG #89 regarding these programs.
Also, note the abridged reprint of "tableofss2graphi" in the article section of this issue.

```
12   " Midgetmenu2.3/c" prg  fast, Mastermenu compatible    37   "Master2.3.ins"    seq  for people who like indexes
68   " Mastermenu v2.3" prg  the disk management utility    39   "syntax article"   seq  for people who like grammer
10   "basic compare.mg" prg  see if 2 progs are the same    47   "tableofss2graphi" seq  for people who like to draw
61   "basic compare.gb" prg   this is same but different    18   "program review"   seq  to review or not to review?
6    "checkbook v3.3"   prg  your computer will itemize,    4    "dekker1.sample"   prg  mutual exclusion algorithim
185  "%checkbook v3.3"  prg   reconcile, sort, edit, ...    4    "dekker2.sample"   prg   for 2 concurrent processes
11   "compare dir's.mg" prg  2-way directory comparisons    1    "f-key saver.mg"   seq  save the f-keys efficiently
18   "compressor.mg"    prg  bigger progs are not better    6    "alarm clock.mg"   prg  rest your eyes five minutes
62   "compressor.mg.gb" prg   but compiled sure is quick    1    "+alrm clck.f0400  prg   after each hour on the VDT
3    "day from date.mg" prg  the Gregorian number system    2    "underflow"        prg  division and multiplication
32   "optimal value.mg" prg  for people who like profits    14   "Comments.mg"      seq  about programs on this disk
33   "checkbook3.3.ins" seq  for people who like cheques
```

GOLD COAST GANDER CBUG #91 NEW RELEASE 2813/14 #13137

By: Fred Peterson, esq.

This is the fourth Goldcoast submission - this time in the form of two (2) related disks.

The disks are identified as "Master" and "Support" disks.

The "Master" disk, as its name implies, includes a reference to _all_ programs.

The "Support" disk was found necessary to run those programs which must use the 8032 emulator mode.

These latter programs are all identified by an exclamation point (!) as a part of each program name. The "master" disk carries each of them for "Background and Instructions" or for "Information" and specific guidance is included with each such reference regarding the procedure which must be carefully followed to "run" them.

Nearly all these programs were initially distributed by a User Group known as "Cursor" during the late 1970's and early 1980's. Although they remain copyrighted material, permission has been obtained for CBUG use.

Others included are believed to be "public domain" material.

Specific reference is made to one program which I have named "mini 2 col dir".

This is a modification of the old "2 column dir prt" or "dir 2 column ptr" program and it permits a miniature directory of two columns to be printed and affixed to the disk jacket for ready reference of all programs on each disk. No need to ever again have paper printouts of all those programs that become so hard to find when needed!

There is still a great deal of material available to be reviewed and adapted for 8050 drive and B-128 computer use.

As time permits, I hooe to continue these conversions as I find I also learn much from this work and so many of the programs are instructive and useful for every day help or are just plain "fun" games!

This is all material that was initially written for the various and sundry "Pet" machines and should be convertible for Cbug use.

Some of the "hurdles" still to be overcome are solving the programs which contain many "pokes and peeks" - not to mention those which are written almost entirely in "machine language".

Well, it's a "labor of love" or something like that - I would be much encoraged to hear from even one of you out there about the wisdom of continuing this conversion as it does take many hours to complete.

Now, here's the annotated directory:

NOTE: Those titles ending in an exclaimation point "!" must be run in conjunction with the second (Support) disk.

0	"golddisk "	04 2c	disk name
6	"introduction"	prg	annimated initial screen.
2	" loader"	prg	controls loading of the following introductory screens and the " menu".
8	" bootscrn .scn"	prg	the introductory screens.
8	" instscrn .scn"	prg	ditto
8	" instscrn2 .scn"	prg	ditto
6	" menu"	prg	control for all "prg" files.
3	" directory "	seq	assist to " menu" file.
19	"mini 2 col dir"	prg	prints a miniature directory to fit on a disk jacket.
7	"pick a card"	prg	just like the magicians do it.
8	"draw poker"	prg	one of the "vegas" machines.
2	"correct time"	prg	a "mini" digital clock.
10	"books of bible"	prg	can you name 'em all?
25	"market"	prg	competition between two companies selling the same product.
16	"bible quiz"	prg	how well do you know your Bible? Select either "fill in" or multiple choice".
19	"maze"	prg	escape, if you can!
20	"hurkle"	prg	for kids (of all ages) - find the "hurkle" hiding in a number field.
30	"bet high card"	prg	play against your "buddies" - with pictures!
33	"rocks"	prg	an African game.
9	"big time"	prg	a B I G watch!
26	"mad"	prg	no - this is not "mothers against drugs" it concocts wild and crazy stories.
10	"estimate"	prg	guess the length or area.
28	"mind"	prg	guess the color pattern.
17	"hman"	prg	NOT "hangman" - just a good word guessing game, without the "noose".
29	"chisanbop"	prg	the Korean method of counting to 99 by using only finger values.
25	"blackjack"	prg	the old "Vegas" standby.
23	"funny faces"	prg	be a police artist - draw all kinds of faces.
60	"miser"	prg	an adventure game with good instructions.
4	"skeet!"	prg	instructions for hitting the birdie
18	"drone!"	prg	instructions for flying a pilotless plane
6	"duel!"	prg	instructions for this dice game
7	"voz!"	prg	instructions for this chess piece identification game
18	"stop!"	prg	instructions for this stop sign capture game
5	"blasto!"	prg	instructions on how to knock out the mines

CBUG WANT ADS

Want ads must be submitted prepaid in legible form (preferably typed or computer printed) atleast 5 weeks before scheduled publication dates. Rates are $10.00 per 80 character/space line. We reserve the right to reformat ads to accomodate printing/layout requirements. Ads must be related to B128 or associated products. Ads for "wanted" or "for sale" are accepted. Display rates on request.

FOR SALE

1.) Three Protecto B128 Systems, $600 each including software. Martha 203-268-7678 or Jim 813-228-0981

2.) B128, 8250, 8023P, Monitor, SSII, SB, Business Software, CBUG Utilities. $450/offer. 509-345-2455 D. Gorsuch, Bx 123, Wilson Creek, Wa. 98860.

3.) FOR SALE (one lot only): B128 lo w/256K, 24K cart, 8050, 4023+ribbon, mon ST10C hard disk, IEEE/Cent (CSI), IEEE/biserial (TNW-2000), all doc, all CABS/Knight/Casey/Deal/Norm/Util/ Games/TC, CR/WR, SSII/SB, all CBUG. C. Chambers, 4220 Dandridge, Alex, Va $700. 703-780-4912. FOB Alex, VA.

4.) B128 system w/ 8050, 4023, amber monitor complete + software. $400 plus shipping. Ed Repic, 2229 Nyon Ave, Anaheim, Ca. 92806 714-520-0778.

5.) B128 computer, 8050 dual disk drive, 8023 heavy duty dot matrix printer, NAP/CEC (80x25) #BM7552 (green) monitor. All cables, manuals, reference guide etc. Software: SS2, SB1, Liz's, CABS SET - A/P, A/R, G/L, P/R, Swan's Util (CBUG #2), Terminal Programs (CBUG #11), Superbase Tutorial/aids (CBUG #13), Financial Utilities (CBUG #5), Norm's Utility (RR#1), Games, Physical Exam 8050, Friendfam (CBUG #15, Superbase-The Book, Issues of THE ESCAPE since 1986. $700 or B.O. John Sullivan, 700 Kentland Dr. Great Falls, 22066-1045. 703-444-2150 evenings.

6.) Complete B128 System: B128, 8050 Dual Floppy Drive, 4023 Printer, Programmers Ref. Guide, Superbase, Superscript II, Basic Tutor Software, User's Guide, Modem Terminal Package, Manuals, All Cables, Original Packing. 203-653-3582 (leave message), Kara Raabe, P.O. Box 165, E. Hartland, Ct. 06027

7.) Complete B128 system. CPU, Monitor, 8050 Drive, 4023 Printer, Anderson Cart. Software: Calc Result, Superbase, Superscript 3, Knights Copy, Physical Exam, Etc. $500 + Frt. Gack Gerhold 517-723-8286

8.) B128, 4023, 8250, monitor, SS, SB $500 or B.O. ST10C 10 Meg Hard Drive (serial or parallel) $500 or B.O. Spare B128 $50. Rich Reichenbach, 1719 N. Cotner, Lincoln, Ne. 68505 402-464-1719

9.) Complete Protecto B128 system plus Superscript, Superbase, Knight's Utility, other software, reference guide, $500 plus shipping. Original manuals and cartons. Ira Bryant. 405-942-0234 days, 405-722-0094 evening.

10.) Protecto B128 w/ 256K, 8050, 4023, Amber Monitor, SS, SSB, CR. Like New. $550 + Shipping. 415-571-7780 Cal.

11.) TWO B128 Commodore Computers, one upgraded to 256K, 1 8050 Dual Disk Drive, 1 4023 100 CPS Bi-directonal Printer. Books, Disks, and other related items. $800 or B.O. plus shipping. Lealia M. Dieck, 1960 Noche Buena St., Seaside, Ca. 93955. 408-394-5191.

12.) Commodore 8032 Computer, 8050 disk drive, 4023 printer, cables. Word Pro 4+, VisiCalc, database, utilities, more. $300. Art Chick, 916-273-9662

13.) Standard B128 system with amber mon, printer, cabling, SS. Any reasonable offer + shipping. C. Lane 714-534-3485

14.) B128 lo-boy computer $90, 8050 drives MPI or Micropolis $200, 4023 printers $75, new boxed prinhead $50, IEEE cables $12 ea, CSI IEEE-Centronics interface $12. Physical Exam $15, SSII $10, SB $10, Superbae the Book $7, Memorex 15" monitor $20. All include shipping in cont. US. 816-942-3615 9AM-9PM.

15.) B128, 8050, 4023, amber monitor, Superscript III, Superbase II, Superbase The Book, Programmer's Reference, misc. Roger Henke, 128 N. Fifth St., Osborne, Ks. 67473 913-346-5504

16.) 8050 Disk Drives $400. 312-456-8720 evenings.

SERVICE CENTER:

TYCOM INC
503 East Street
Pittsfield, MA 01201
(413) 442-9771

Authorized Commodore service center, we repair B128 8032, 4023, 8023, 6400, 4040, 8050, 8250, 2031, C64 C-128, etc. Rates are $50/Hr. plus parts. We will diagnose & quote repair cost for $25. Normal turn-around time is 5 business days. We ship via UPS. Limited supply 8032's and 4023's (new) at $189 each

Wanted:
NEW HOME For Some Of The Best Available Commodore B Series Computer Equipment

Systems Are Pictured on Page 26, Fall, 1986 (Fifth Issue) of *The CBUG Escape*

Dear CBUG Members; Sadly, I must (for personal reasons) sell most of my beloved Commodore "B" computers, peripherals and software. Here is what is available. Price and quantity listed. Shipping extra.

1. B-128 Low Profile Computer..$95 ea. (6 available)..3 never used.
2. 8050 disk-drive ... $395 ea. (5 avail.)
3. 4023 dot matrix printer... $95 ea. (2)
4. 8250 disk drive ... $475 ea. (1 only)
5. 8023 dot matrix printer ... $185 ea. (1)
6. 6400 daisy printer ... $395 ea. (1)
7. Rare B-256-80 High Profile Computer with swivel monitor and detachable keyboard (best 8-bit business computer ever made), with high res screen. Perfect for word processing ... $345 ea. (3)
8. Calc Result Spreadsheet .. $50 ea. (3)
9. Word Result Wordprocessor .. $50 (1)
10. Book ... *Power of Calc Result* .. $7 ea. Postpaid (45 copies) Best book for those with Calc Result - ten examples.
11. Zenith (Protecto monitor) ... $40 (4)
12. Miscellaneous cables, software, ADA 1800 Parallel Interface, ADA 1450 Serial Interface, and more. (Call for prices).

Write: Fred Lovejoy, 2101 North 69th Place, Scottsdale, Arizona 85257 ...
Or Call: (602) 946-0202 (5-9 PM Mtn.)

-- BeeLine V2.1 -- BeeLine V2.1 --

>> THE PROFESSIONAL TERMINAL PROGRAM FOR THE B-128 COMPUTER <<

LOAD/SAVE in ASCII or CBM Format
Skip beginning of large files

VIEW memory Buffers using forward and backward controls
Set Blocks for Saving & Printing
Transfer Data between Buffers
Find strings within Buffer Data

PRINT to CBM or ASCII Printers
Use various device numbers

RECEIVE directly to Disk files
Up to 500K using CBM 8050 Drive
Use with/without Memory Buffers

MESSAGE Editing within BeeLine
Use 9 seperate Messages with screen editing & ESC codes
Transmit & Edit using Fkeys
Prepare Message from Buffer, previous or blank Message

XMODEM Upload and Download
Exchange any Disk file
Visually monitor transfers

TRANSMIT Data from Messages, any data file or Buffer
Delay after char. or line
Use XON/XOFF flow control

AVAILABLE IMMEDIATELY FOR ONLY $40.00 - ORDER FROM CBUG TODAY!

BeeLine V2.1 is Copyrighted 1986 by J & K Lemkelde
LemData Software, P.O. Box 175, Dover, PA 17315

!!! GET YOURSELF A POWERHOUSE !!!

B-1024 1 MEGABYTE MEMORY EXPANSION CIRCUIT BOARD

* In the Low Profile B-128, Implements Banks 0 through 14 with 256K by 1, 150ns socketed Dynamic Memory. This increase provides 8 Documents with SuperScript II; 28,560 Input Values with Calc Result; and 14 8032s with the 8432 Emulator!

* Fully Assembled and Tested. Plugs internally onto pin fields in the low profile B-128. You can install in minutes at your site! The original configuration of a stock B-128 can be restored at any time by simply removing the board. This is a non-destructive upgrade!

* Includes Pin Fields for future I/O, RAM Socket for $0800 to $1FFF in Bank 15, Documentation for Installation, Parts Layout, Schematics, and a Machine Language DRAM Memory Test Utility on 8050 Diskette!

* Three Memory Density Options available to fit your applications! You can purchase the 41256-15 DRAM in the market place at your convienence to fully populate the board!

24K RAM/ROM CARTRIDGE For The B-128/CBM-256

* Increase Memory! Add Another 24K of Memory to your System in BANK 15 from $2000-$7FFF! Complete! Assembled and Tested! Plugs into the Cartridge Port on the back of your main system unit!

* Run Scott's B-MON, Serial Bus Software, Custom Machine Language Applications, Keytrix, JCL Work Shop, Harrison's Assembler, and More!

RAM/ROM SOCKET For $0800-$1FFF

* Implements the 6K memory area below the Cartridge Port in BANK 15! A small Circuit Board with Connector and Socket that will receive a 6264LP-15 SRAM or a 2764-25 UVPROM to customize your application!

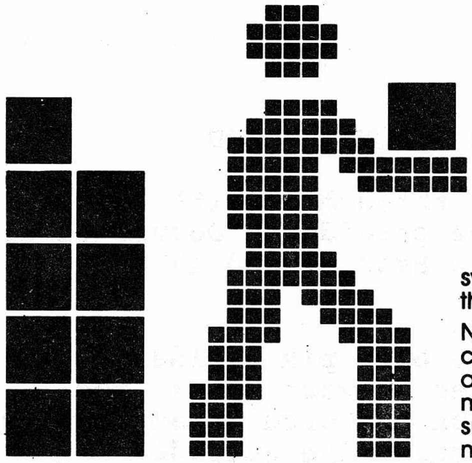
COMMODORE SERIAL BUS INTERFACE for the B-128

* A Full Featured Hardware Interface for the B-128 Implementing the Commodore Serial Bus with the Functions of Controller, Listener/Talker, Slow Bus, Fast Bus, Attention Acknowledge, Power-On Serial Bus Reset, Manual Serial Bus Reset! Comes in Rugged Plastic case! $ 59.95

B-1024 1 MEGABYTE MEMORY EXPANSION BOARD for the Low Profile B-128!

	B-1024 with 1024K Memory Installed	$699.00
ANDERSON	B-1024 with 512K Memory Installed	$499.00
COMMUNICATIONS ENGINEERING	B-1024 with 0K Memory Installed	$289.00
2560 Glass Rd. NE.	24K RAM/ROM CARTRIDGE with Case	$ 84.95
Cedar Rapids, Iowa		
U.S.A. 52402	RAM/ROM Socket for $0800-$1FFF in BANK 15	$ 24.95

TERMS: FREE SHIPPING IN USA, USA FUNDS, IOWA ADD 4%, ALLOW 6-8 WEEKS



NWM'S INVENTORY CONTROL SYSTEM* VERSION 1.3

This system was developed by a user for users. It seems that most systems are developed by computer engineers that never see or use the product when finished.

NWM, inc. chose the toughest software analyst we could find to give a complete and unbiased review. Bob Loefler is the gentleman that tore apart the CABS Accounting System and documented all the bugs in most of the modules. To have someone of this caliber make suggestions means better, more efficient software available for CBUG members.

- Loads program modules in less than **8 seconds** (superbase 2) to main menus in **3 seconds** or less
- On screen **pop-up calculator** in transaction modules
- Most data centered functions use the **calculator keypad**
- Versatile report features allow for **3 ways** to print the same report. User selects the fastest method
- Built in sophisticated export program allows for **complete packing** of the database
- Type ahead feature allowed
- You can display reports on screen
- Access to superbase menu for user developed applications
- Partial-match key search now allowed in transaction modules
- User can search forward and backward through file while in transaction modules
- Elimination of most tiresome prompts for faster movements between subfunctions and menus
- Simplification of entering prices and costs that have not changed when entering transactions
- User programmable calculator allows you to turn the calculator function on or off within the transaction modules. If the user does not need the calculator, it saves time by eliminating the prompt and bypasses the calculator function

NWM, inc. proudly presents Inventory Control 2.0 w/dual key indexing only $^{\$}49^{95}$

Listen to how NWM has improved this already great Inventory Control System...

- Dual indexing (allows user to select record by item or stock number)
- New records can be entered in issues, receipts and orders modules
- Price list report (w/o cost for customer viewing)
- Sales report (sales history by item per month for 12 months)
- Cost summary (calculates total cost for a category of items selected)
- Configuration module (allows user friendly configuration of printers, drives etc.)

Only a $19.95 upgrade charge for owners of previous versions of NWM Inventory.

WE SUPPORT OUR CUSTOMERS!

NWM Inventory Control System Version 1.3 Prices

B Version 1 8050	**$39.95**	C-128 Version 1 1571	**$24.95**
B Version 2 8050	**$39.95**	B-128 Version 1&2 8050...	**$44.95**

U.S. shipping and handling charge $3.95 (Prepaid)

NWM, inc. • 539 N. Wolf Rd. •Wheeling, IL 60090

(312) 520-2540 Hours— (Voice) **Mon.-Thur. 12:30-5:00, Sat. 12:00-4:00** (24 Hour Order Recorder)

Superbase is a registered trademark of Precidion Software.

B128 and C128 are registered trademarks of Commodore Business Machines Limited.

*Requires use of Superbase®

CBM 128-80 w/8088 Co-processor

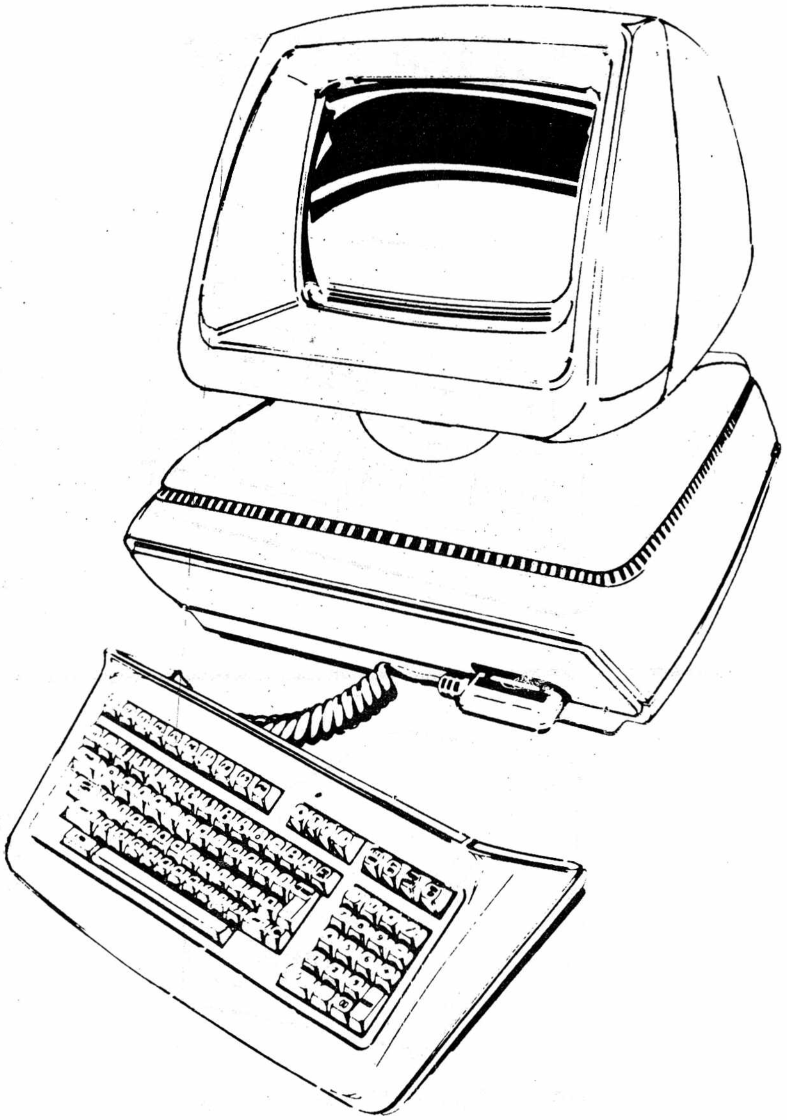
Call for Details
Very Limited Quantities

We are offering a limited number of used CBMX 128-80 computers for only $325. These come installed and tested with the original 8088 co-processor boards that until recently were hidden in the Commodore research labs.

The CP/M 86* operating system was implemented on the B system and has been tested and found to be very reliable. It had previously been stated that the co-processor would only work with the CBM 256-80. This is not true! The PLA that is installed on the hi-profile motherboard determines whether the operating system will run on that particular machine. We will also be offering the 8088 co-processor board for $80 providing the purchaser sends in their working hi-profile motherboard with the correct PLA. Call for details!

All the generic CP/M 86 software that we have tested will operate on the B series machine with an 8050 drive. The SFD and the 8250 can also be used with some restrictions. These same programs can also be run under the Digital Research CP/M 86 and Concurrent PC Dos operating system on an IBM. Therefore an investment in software is not wastec as it can be ported to other compatible computers.

*CP/M 86 is a trademark of DRI Inc.



In an effort to promote the CP/M-86 and Ms-Dos project, N.W. Music has decided to sell the remaining co-processor boards at our cost to help stimulate this effort. The price will be $80 and purchase will be subject to certain stipulations. It had been previously stated, that these boards would not work with the lo-profile computer. This does not seem to be the case. It has been suggested by a prominent member of the group that any problems could be only power supply related. We have a lo-profile model up and running. There are only a few boards remaining so call for details.

Co-processor $80

CBM 128 & 256 features:

- SWIVEL MONITOR
- ADJUSTS HOR. and VERT.
- DETACHABLE KEYBOARD
- 9 x 14 PIXEL DISPLAY
- INCREDIBLE RESOLUTION
- DESIGNED FOR 2 INTERNAL DRIVES

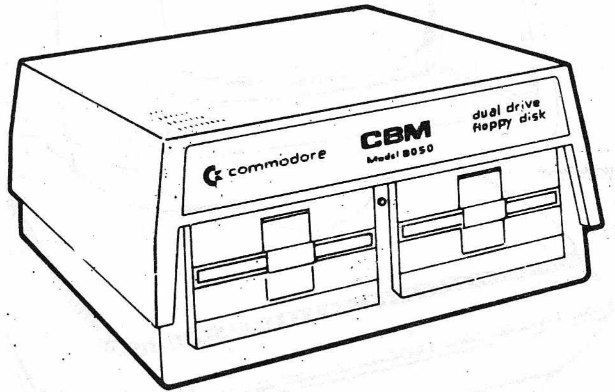
Priced from $199 to $350 (US)
SHIPPING CHARGES EXTRA!

You really have to see the green phosphor display to believe it. This model has at least as good a display as the other company with those three big letters.

NORTH WEST MUSIC CENTER INC.

539 N. Wolf Rd. — Wheeling, IL 60090
Hours— (Voice) Phone (312) 520-2540
Mon.-Thur. 12:30-5:00, Sat. 12:00-4:00
(24 Hour Order Recorder)

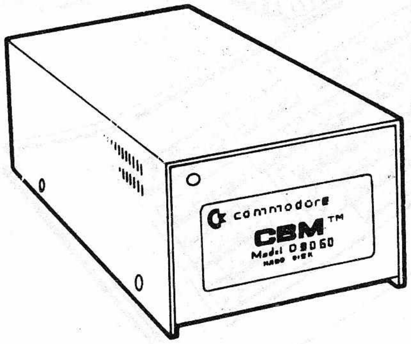
CBM DISK DRIVE CORNER



8050 Dual Disk Drive

The model 8050 dual floppy disk unit uses a 100 Track Per Inch (TPI) single headed drive with a storage capacity of 533,248 bytes per drive. Each 8050 diskette has 77 tracks, and is read/write compatible with the model 8250 disk drive. This compatibility is limited to one side of the diskette.

NEW PRICED AT **$400.00** (U.S.)
USED or REHABS FROM **$225.00** when available
ADD $16.95 SHIPPING (U.S.)



REHABS PRICED AT **$495.00** (U.S.)
ADD $15.95 SHIPPING (U.S.)

D9060 & D9090 Hard Disk Drive

The two models of hard disk units are the 5¼" single-drive non-removable "Winchester" technology storage devices. The D9060 and D9090 units feature two or three platters with recording surfaces on both sides and provide respectively 5.0 or 7.5 million characters of storage. A single random access file may occupy the entire capacity of either unit. An IEEE Interface connector is located on the back of the drive. Near the lower edge of the rear panel is a "slow blow" fuse, and an AC power cord.

SFD 1001 1 Megabyte Drive double sided 8250 format IEEE interface

N.W. Music now offers a constructive upgrade for the SFD-1001+ increasing its function and versatility. This modification allows the user to operate in a true single drive 8050 mode. This eliminates the previous problem of loading most 8050 software. This upgrade is only $19.95 over the base SFD price. This modification allows the user to select, by switch, which mode they desire to operate during power down. The user will stay in this mode until power down and switch reversal.

PRICED AT **$149.95** (U.S.)
ADD $10.45 SHIPPING (U.S.)
PRICED AT **$169.90** (U.S.) SFD with 8050 switch
ADD $10.45 SHIPPING (U.S.)

NORTHWEST MUSIC CENTER INC.

539 N. Wolf Rd. — Wheeling, IL 60090
Hours— (Voice) Phone (312) 520-2540
Mon.-Thur. 12:30-5:00, Sat. 12:00-4:00
(24 Hour Order Recorder)

INVENTORY NUMBER CROSS INDEX

Spring 1988

INV #	ITEM ID	PRICE	DESCRIPTION
11221	.I to c	$ 35.00	COMMODORE IEEE to Centronics Adaptor Board (6400 type)
11236	.TYPE CHG	$ 15.00	Changes connector type above to Centronics
11330	.ZENERS	$ 6.00	Close tollerance Zener diodes & instructions for 8050 /pair
11344	.P-DIODES	$ 8.50	Precision Reference diodes & instructions for 8050 repair /p
11536	CBUG #32	$ 10.00	Kernaghan's Utilities v3
11540	CBUG #33	$ 9.00	Medical Accounting (SUPERBASE application)
11555	VOL 8	$ 4.00	Summer 1987 ESCAPE, copy of publication
11593	CBUG #71	$ 25.00	Precision Church Accounting(Superbase application)
11606	CBUG #70	$ 9.00	Summer 1987 ESCAPE print files - disk
11611	PR8	$ 9.00	CPM 86 Info & Programs #1
11625	PR9	$ 9.00	CPM 86 Info & Programs #2
11630	PR10	$ 9.00	CPM 86 Info & Programs #3
11659	CBUG #60	$ 16.00	Liz Deal's Took Kit (Utilities)(upgrade)
11697	CBUG #62	$ 9.00	Super Church (SUPERBASE application)
11709	CBUG #63	$ 30.00	The New King James New Testament (on 2 disks)
11714	CBUG #64	$ 9.00	Sermons 2
11728	CBUG #65	$ 9.00	Sermons 3
11733	CBUG #66	$ 11.00	The NEW 8432 Emulator v.g & More
11747	CBUG #67	$ 9.00	** B128 Kernal/editor Source Code
11752	CBUG #68	$ 9.00	** Basic Source Code B128 & Others
11766	CBUG #69	$ 9.00	Basic Source Code under study (Brezinski)
11771	CBUG #40	$ 10.00	Public Domain Math A
11785	CBUG #41	$ 10.00	Public Domain English A
11790	CBUG #42	$ 10.00	Public Domain GHBT
11803	CBUG #43	$ 10.00	Public Domain Science A
11818	CBUG #44	$ 10.00	Public Domain Science B
11822	SET of 5	$ 45.00	Public Domain CBUG #40 thr #44 inclusive
11837	CBUG M45	$ 9.00	CBUG Utilities & Misc. #3 (mislabeled #2 fall 87)
11856	CBUG #47	$ 16.00	dFile database pgm--Available ONLY to US members
11860	CBUG #48	$ 9.00	CBM Diagnostics adapted for the B128
11875	CBUG #49	$ 9.00	Medical Finance #2 (SUPERBASE application)
11894	CBUG #51	$ 29.00	JCL Work Shop & Assembler -- 2 disk set
11906	CBUG #52	$ 9.00	Summer part 2 1986 ESCAPE print files - disk
11911	CBUG #53	$ 9.00	Fall 1986 ESCAPE print files - disk
11925	CBUG #58	$ 12.00	Dittinger's Utilities +
11930	CBUG M54	$ 9.00	CBUG Misc. #4
11944	CBUG M55	$ 9.00	ML Programming Information
11959	CBUG #56	$ 35.00	Harrison's Assembler, revised. 5.5 v8
11963	CBUG #57	$ 9.00	Goceliak Strikes Again
11978	CBUG #59	$ 9.00	Winter/Spring 1987 ESCAPE print files - disk
11982	CBUG #72	$ 14.00	Goceliak's Third Mine (Gold that is)
11997	CBUG #73	$ 14.00	THE ESCAPE Index (Superbase app) v.Fall 87
12007	CBUG #74	$ 15.00	SuperOffice Scrubber
12012	CBUG #75	$ 9.00	Machine Language Index (Superbase application)
12026	CBUG #76	$ 40.00	The King James Bible, complete. Set of 9 disks
12031	CBUG #77	$ 20.00	Super Teacher - High Capacity. 4 disk set + blanks
12045	CBUG #78	$ 9.00	Gold Coast Instructional
12050	PR #15	$ 9.00	Dave Wack's Assortment
12064	CBUG M80	$ 9.00	CBUG MISC 12-87
12079	CBUG #81	$ 9.00	David Green's Update
12083	PR #11	$ 9.00	CP/M 86 info & Pgms #4
12098	PR #12	$ 9.00	CP/M 86 info & Pgms #5
12101	PR #13	$ 9.00	CP/M 86 info & Pgms #6
12116	PR #14	$ 9.00	CP/M 86 info & Pgms #7
12168	CBUG #79	$ 9.00	Fall 1987 ESCAPE print files - disk
12173	VOL 7	$ 7.00	Winter Spring 1987 ESCAPE, copy of publication
12187	VOL 9	$ 6.00	Fall 1987 ESCAPE, copy of publication

INV #	ITEM ID	PRICE	DESCRIPTION
12192	PE 8250	$ 35.00	PHYSICAL EXAM for the 8250 and SFD-1001 Disk Drives
12204	KNIGHT'S	$ 20.00	8050 (DOS 2.7) COPY UTILITY
12219	PE 8050	$ 35.00	PHYSICAL EXAM for the 8050 Disk Drive
12223	PE 1541	$ 35.00	Physical Exam for the 1541
12238	PE 4040	$ 35.00	Physical Exam for the 4040
12242	PE 1571	$ 35.00	Physical Exam for the 1571
12257	CBUG #11	$ 14.00	Terminal Pgms w/ BTerm
12261	CBUG #11a	$ 9.00	Terminal Pgms w/o BTerm
12280	BEELINE	$ 40.00	v2.1 Telecommunications Program
12295	9060/9090	$ 25.00	Service Manual - Photocopy (allow extra week)
12308	SFD-1001	$ 5.00	Schematics by photocopy & stat reductions
12332	VOL 10	$ 6.00	Winter 1988 ESCAPE, copy of publication
12346	VOL 6	$ 5.00	Fall 1986 ESCAPE, copy of publication
12449	VOL 2 & 3	$ 6.00	Winter/Spring 1986 ESCAPE, copy of publication
12468	VOL 4	$ 4.00	Summer 1986 ESCAPE Part 1, copy ofpublication
12473	VOL 5	$ 3.00	Summer 1986 ESCAPE Part 2, copy of publication
12492	CBUG #27	$ 9.00	Goceliaks Gold Mine - disk utilities/engineering
12504	CBUG #28	$ 19.00	Casey's Scrubber
12519	CBUG #29	$ 9.00	CBUG TPUG P1 & P2
12538	CBUG #31	$ 9.00	Superbase Corner & Hints
12542	PR5	$ 9.00	Pre Release #5
12557	PR6	$ 6.00	Pre Release #6 partial
12561	CBUG #36	$ 9.00	London Sampler
12576	CBUG #37	$ 19.00	SUPERPRINT collection
12580	CBUG #38	$ 9.00	Summer part 1 1986 ESCAPE print files - disk
12608	.SER.CART	$ 65.00	SERIAL BUS CARTRIDGE ADAPTOR (Anderson)
12613	.24K RAM	$ 85.00	24K RAM/ROM Cartridge (Anderson)
12651	CBUG #26	$ 9.00	Jan. 1986 Telecom issue and CBUG #25 overflow
12665	CBUG #25	$ 9.00	Winter/Spring 1986 ESCAPE print files - disk
12699	CBUG #21	$ 9.00	Retail News Distribution pgm
12701	CBUG #22	$ 9.00	Math Education Programs
12716	CBUG #15	$ 14.00	Friendfam (SUPERBASE application pgm)
12720	CBUG #24	$ 9.00	8432 Emulator Disassembled
12735	CBUG #23	$ 15.00	Bible Games
12749	PR4	$ 9.00	Pre Release #4
12768	CBUG M20	$ 9.00	CBUG Utilities etc #2
12773	CBUG #16	$ 19.00	Swan's Basic Course
12787	CBUG #13	$ 9.00	SUPERBASE tutorial pgms & Leighfield aids texts
12792	CBUG #18	$ 10.00	Games and Education
12805	CBUG #19	$ 9.00	Old BUG texts and programs
12824	PR1	$ 9.00	Pre Release #1
12839	PR2	$ 9.00	Pre Release #2
12843	PR3	$ 9.00	Pre Release #3
12862	CBUG # 1	$ 9.00	Norm's Utility v1.2
12881	CBUG # 3	$ 14.00	Swan's Utility #1
12913	CBUG # 6	$ 9.00	CBUG/TPUG #1
12932	CBUG # 7	$ 9.00	Northrup's SUPERBASE Applications
12946	CBUG # 8	$ 9.00	Sermons 1
12951	CBUG # 9	$ 9.00	CABS GL pro forma #1
12965	CBUG #10	$ 9.00	Fall 1985 ESCAPE and prior files - disk
12984	CBUG #12	$ 14.00	Scott's B-Mon
13014	CBUG #M82	$ 9.00	CBUG MISC. 0588
13028	CBUG #83	$ 14.00	Goceliak's Goldmine #0588
13033	CBUG #84	$ 35.00	FAST BUS Programs Disk (Jarvis/Springer)
13047	CBUG #85	$ 9.00	Gold Coast Gaggle
13052	CBUG #86	$ 14.00	Gold Coast Tutorial
13066	PR #16	$ 9.00	CP/M 86 2.001
13071	PR #17	$ 9.00	CP/M 86 2.002
13085	CBUG #87	$ 9.00	Winter 1988 ESCAPE print files - disk

USE PHOTOCOPIES FOR MULTIPLE ORDERS - ADDITIONAL ORDER FORMS WILL BE INCLUDED WITH SHIPMENT

Spring 88

NAME ______________________________

ADDRESS ______________________________

CITY ____________________ STATE ______ ZIP __________

THIS IS YOUR SHIPPING LABEL -- PRINT OR TYPE NEATLY

ORDERS OVER 4 DISKS OR PHYSICAL EXAM SHIPPED BY BY UPS WITHIN CONTINENTAL U.S. YOU MUST USE STREET ADDRESSES FOR SUCH ORDERS -- UPS CAN NOT DELIVER TO POST OFFICE BOXES!

Your phone # ______________________________

Remit to: CBUG, Inc., 4102 N. Odell, Norridge, Il. 60634 U.S.A. All payments must be in US funds.
IMPORTANT: If your shipping address is different than your membership address, please give your membership address!!

My Membership Address (if different) is: ____________ ____________ ______ ______
street city state zip

R ck mo ca pr S

Description	Quantity	Stock #	Price	Extension
COMMODORE IEEE to Centronics Adaptor Board (6400 type)	______	11221	35.00	______
CBUG connector type changer for above (changes output connector to standard Centronics "blue ribbon" D shell type	______	11236	15.00	______
*SERIAL BUS CARTRIDGE ADAPTOR (Anderson) (use with CBUG #84 Fast Bus programs)	______	12608	65.00	______
*24K RAM/ROM Cartridge (Anderson)	______	13033	85.00	______
BALANCE FWD FROM DISK & DATA CASE ORDER FORMS, pages A11 & 12 --->				______
<North America addresses only!!!> Illinois residents add 7% Sales tax to above items only				______
CBUG #88 Goceliak's Goldmine #0788	______	13118	19.00	______
CBUG #84 FAST BUS Programs Disk (Jarvis/Springer) . . .	______	13033	35.00	______
CBUG #89 SYNTAX CHECKER & Misc	______	13118	26.00	______
CBUG #90 Mathew Goldstein's Assortment 0988	______	13122	9.00	______
CBUG #91 Gold Coast Gander - Two disk set	______	13137	10.00	______
CBUG #92 Restaurant Accounting	______	13156	9.00	______
CBUG #93 B-128 Service Information	______	13160	9.00	______
PR #18 CP/M 86 2.003	______	13103	9.00	______
CBUG #94 Spring 1988 ESCAPE Print File	______	13103	9.00	______
______________________	______	______	______	______
______________________	______	______	______	______
KNIGHT's 8050 (DOS 2.7) COPY UTILITY	______	12204	20.00	______
PHYSICAL EXAM for the 8050 Disk Drive	______	12219	35.00	______
PHYSICAL EXAM for the 8250 and SFD-1001 Disk Drives . .	______	12192	35.00	______
BEELINE v2.1 Telecommunications Program	______	12280	40.00	______
Close tollerance Zener diodes & instructions for 8050 $6/pair	______	11330	6.00	______
Precision Reference diodes & instructions for 8050 repair /pair	______	11344	8.50	______
Recopy fee remittance	______	12401	5.00	______
BALANCE FORWARD FROM ADDITIONAL ORDER FORM			------->	______
BALANCE FORWARD FROM ADDITIONAL ORDER FORM			------->	______
BALANCE FORWARD FROM ADDITIONAL ORDER FORM			------->	______
MERCHANDISE SHIPPING AND HANDLING CHARGE --- ALWAYS INCLUDED, NO EXCEPTIONS				$ 2 . 0 0
	SUBTOTAL		$	______
Free Will Contribution to CBUG		12416		______
Extra Copy of this issue, Spring 1988 CBUG ESCAPE	______	13189	6.00	______

BY CHECK ___ BY MONEY ORDER ___ TOTAL REMITTED $ __________

* These are special ordered by CBUG as a convenience for members. There is an extra $5.00 handling charge built in to these prices. Pricing and availability subject to change. Please allow extra time for trans-shipping. You may prefer to order directly from Mr. Anderson -- see ad page 2.

NOTE: We endeavor to ship against money order within 5 business days of receipt. Others delayed 14 to 18 days.

Spring 1988

CBUG LIBRARY ORDER FORM #2

ITEM ID	INV #	QUANTITY	EACH	EXTENSION	DESCRIPTION
.24K RAM	12613		$85.00		24K RAM/ROM Cartridge (Anderson)
.I to c	11221		35.00		COMMODORE IEEE to Centronics Adaptor Board (6400)
.P-DIODES	11344		8.50		Precision Reference diodes for 8050 & instructions
.SER.CART	12608		65.00		SERIAL BUS CARTRIDGE ADAPTOR (Anderson)
.TYPE CHG	11236		15.00		Changes connector type above 6400 to Centronics
.ZENERS	11330		6.00		Close tollerance Zener diodes (8050) & instructions
9060/9090	12295		25.00		Service Manual - Photocopy (allow extra week)
BEELINE	12280		40.00		v2.1 Telecommunications Program
CBUG # 1	12862		9.00		Norm's Utility v1.2
CBUG # 3	12881		14.00		Swan's Utility #1
CBUG # 6	12913		9.00		CBUG/TPUG #1
CBUG # 7	12932		9.00		Northrup's SUPERBASE Applications
CBUG # 8	12946		9.00		Sermons 1
CBUG # 9	12951		9.00		CABS GL pro forma #1
CBUG #10	12965		9.00		Fall 1985 ESCAPE and prior files - disk
CBUG #11	12257		14.00		Terminal Pgms w/ BTerm
CBUG #11a	12261		9.00		Terminal Pgms w/o BTerm
CBUG #12	12984		14.00		Scott's B-Mon
CBUG #13	12787		9.00		SUPERBASE tutorial pgms & Leighfield aids texts
CBUG #15	12716		14.00		Friendfam (SUPERBASE application pgm)
CBUG #16	12773		19.00		Swan's Basic Course
CBUG #18	12792		10.00		Games and Education
CBUG #19	12805		9.00		Old BUG texts and programs
CBUG #21	12699		9.00		Retail News Distribution pgm
CBUG #22	12701		9.00		Math Education Programs
CBUG #23	12735		15.00		Bible Games
CBUG #24	12720		9.00		8432 Emulator Disassembled
CBUG #25	12665		9.00		Winter/Spring 1986 ESCAPE print files - disk
CBUG #26	12651		9.00		Jan. 1986 Telecom issue and CBUG #25 overflow
CBUG #27	12492		9.00		Goceliaks Gold Mine - disk utilities/engineering
CBUG #28	12504		19.00		Casey's Scrubber
CBUG #29	12519		9.00		CBUG TPUG P1 & P2
CBUG #31	12538		9.00		Superbase Corner & Hints
CBUG #32	11536		10.00		Kernaghan's Utilities v3
CBUG #33	11540		9.00		Medical Accounting (SUPERBASE application)
CBUG #36	12561		9.00		London Sampler
CBUG #37	12576		19.00		SUPERPRINT collection
CBUG #38	12580		9.00		Summer part 1 1986 ESCAPE print files - disk
CBUG #40	11771		10.00		Public Domain Math A
CBUG #41	11785		10.00		Public Domain English A
CBUG #42	11790		10.00		Public Domain GHBT
CBUG #43	11803		10.00		Public Domain Science A
CBUG #44	11818		10.00		Public Domain Science B
CBUG #47	11856		16.00		dFile database pgm--Available ONLY to US members
CBUG #48	11860		9.00		CBM Diagnostics adapted for the B128
CBUG #49	11875		9.00		Medical Finance #2 (SUPERBASE application)
CBUG #51	11894		29.00		JCL Work Shop & Assembler -- 2 disk set
CBUG #52	11906		9.00		Summer part 2 1986 ESCAPE print files - disk
CBUG #53	11911		9.00		Fall 1986 ESCAPE print files - disk
CBUG #56	11959		35.00		Harrison's Assembler, revised. 5.5 v8
CBUG #57	11963		9.00		Goceliak Strikes Again
CBUG #58	11925		12.00		Dittinger's Utilities +
CBUG #59	11978		9.00		Winter/Spring 1987 ESCAPE print files - disk
CBUG #60	11659		16.00		Liz Deal's Took Kit (Utilities)(upgrade)
CBUG #62	11697		9.00		Super Church (SUPERBASE application)
CBUG #63	11709		30.00		The New King James New Testament (on 2 disks)
CBUG #64	11714		9.00		Sermons 2

=========

PAGE TOTAL __________

Spring 1988

CBUG LIBRARY ORDER FORM #3

CBUG #65	11728		9.00		Sermons 3
CBUG #66	11733		11.00		The NEW 8432 Emulator v.g & More
CBUG #67	11747		9.00		** B128 Kernal/editor Source Code
CBUG #68	11752		9.00		** Basic Source Code B128 & Others
CBUG #69	11766		9.00		Basic Source Code under study (Brezinski)
CBUG #70	11606		9.00		Summer 1987 ESCAPE print files - disk
CBUG #71	11593		25.00		Precision Church Accounting(Superbase application)
CBUG #72	11982		14.00		Goceliak's Third Mine (Gold that is)
CBUG #73	11997		14.00		THE ESCAPE Index (Superbase app) v.Fall 87
CBUG #74	12007		15.00		SuperOffice Scrubber
CBUG #75	12012		9.00		Machine Language Index (Superbase application)
CBUG #76	12026		40.00		The King James Bible, complete. Set of 9 disks
CBUG #77	12031		20.00		Super Teacher - High Capacity. 4 disk set + blanks
CBUG #78	12045		9.00		Gold Coast Instructional
CBUG #79	12168		9.00		Fall 1987 ESCAPE print files - disk
CBUG #81	12079		9.00		David Green's Update
CBUG #83	13028		14.00		Goceliak's Goldmine #0588
CBUG #84	13033		35.00		FAST BUS Programs Disk (Jarvis/Springer)
CBUG #85	13047		9.00		Gold Coast Gaggle
CBUG #86	13052		14.00		Gold Coast Tutorial
CBUG #87	13085		9.00		Winter 1988 ESCAPE print files - disk
CBUG #M82	13014		9.00		CBUG MISC. 0588
CBUG M20	12768		9.00		CBUG Utilities etc #2
CBUG M45	11837		9.00		CBUG Utilities & Misc. #3 (mislabeled #2 fall 87)
CBUG M54	11930		9.00		CBUG Misc. #4
CBUG M55	11944		9.00		ML Programming Information
CBUG M80	12064		9.00		CBUG MISC 12-87
KNIGHT'S	12204		20.00		8050 (DOS 2.7) COPY UTILITY
PE 1541	12223		35.00		Physical Exam for the 1541
PE 1571	12242		35.00		Physical Exam for the 1571
PE 4040	12238		35.00		Physical Exam for the 4040
PE 8050	12219		35.00		PHYSICAL EXAM for the 8050 Disk Drive
PE 8250	12192		35.00		PHYSICAL EXAM for the 8250 and SFD-1001 Disk Drives
PR #11	12083		9.00		CP/M 86 info & Pgms #4
PR #12	12098		9.00		CP/M 86 info & Pgms #5
PR #13	12101		9.00		CP/M 86 info & Pgms #6
PR #14	12116		9.00		CP/M 86 info & Pgms #7
PR #15	12050		9.00		Dave Wack's Assortment
PR #16	13066		9.00		CP/M 86 2.001
PR #17	13071		9.00		CP/M 86 2.002
PR1	12824		9.00		Pre Release #1
PR10	11630		9.00		CPM 86 Info & Programs #3.
PR2	12839		9.00		Pre Release #2
PR3	12843		9.00		Pre Release #3
PR4	12749		9.00		Pre Release #4
PR5	12542		9.00		Pre Release #5
PR6	12557		6.00		Pre Release #6 partial
PR8	11611		9.00		CPM 86 Info & Programs #1
PR9	11625		9.00		CPM 86 Info & Programs #2
SET of 5	11822		45.00		Public Domain CBUG #40 thr #44 inclusive
SFD-1001	12308		5.00		Schematics by photocopy & stat reductions
VOL 10	12332		6.00		Winter 1988 ESCAPE, copy of publication
VOL 2 & 3	12449		6.00		Winter/Spring 1986 ESCAPE, copy of publication
VOL 4	12468		4.00		Summer 1986 ESCAPE Part 1, copy ofpublication
VOL 5	12473		3.00		Summer 1986 ESCAPE Part 2, copy of publication
VOL 6	12346		5.00		Fall 1986 ESCAPE, copy of publication
VOL 7	12173		7.00		Winter Spring 1987 ESCAPE, copy of publication
VOL 8	11555		4.00		Summer 1987 ESCAPE, copy of publication
VOL 9	12187		6.00		Fall 1987 ESCAPE, copy of publication

=========

PAGE TOTAL ________

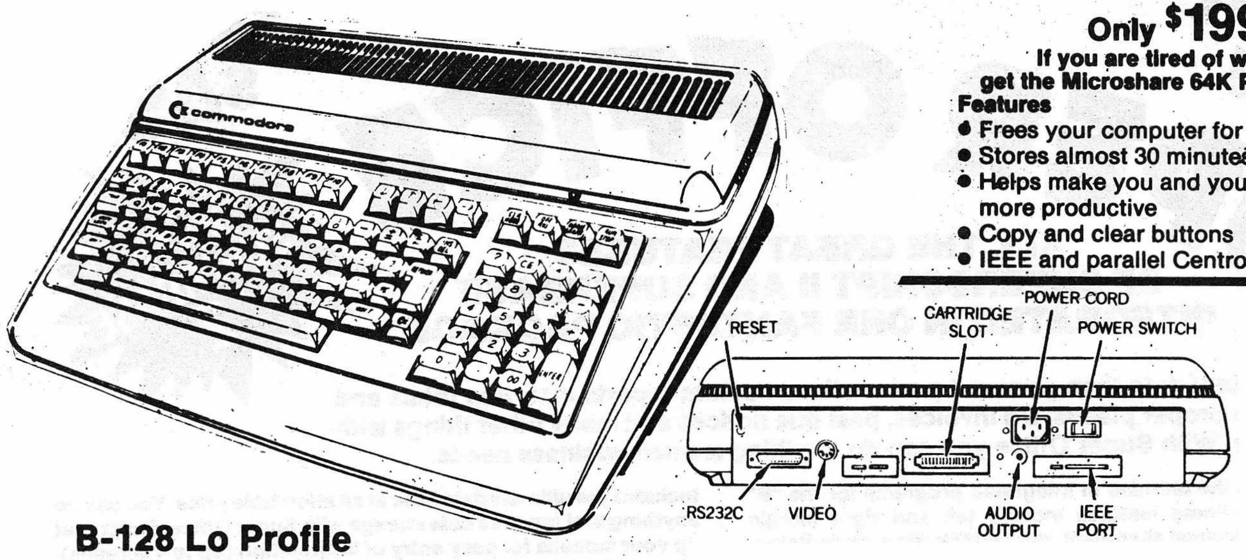
64K IEEE Print Buffer

Only $199

If you are tired of waiting, get the Microshare 64K Print Buffer.

Features

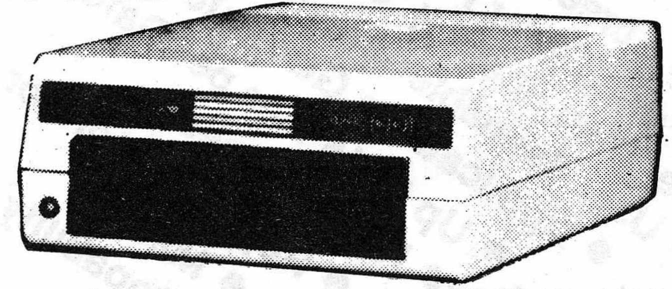
- Frees your computer for other tasks
- Stores almost 30 minutes of printing
- Helps make you and your employees more productive
- Copy and clear buttons
- IEEE and parallel Centronix output



B-128 Lo Profile

We have a very limited supply of new B-128-80 computers left. The available supply will last no longer than late October to early November if current sales hold-up. We also have a supply of Commodore rehabbed machines. If you desire a new machine, please order now! There are no new machines available once our supplies run out.

PRICED AT **$145** new
$99 rehabbed
ADD $11.45 SHIPPING (US)



SFD 1001 1 Megabyte Drive double sided 8250 format IEEE interface

N.W. Music now offers a constructive upgrade tor the SFD-1001+ increasing its function and versatility. This modification allows the user to operate in a true single drive 8050 mode. This eliminates the previous problem of loading most 8050 software. This upgrade is only $19.95 over the base SFD price. This modification allows the user to select, by switch, which mode they desire to operate during power down. The user will stay in this mode until power down and switch reversal.

PRICED AT **$149.95** (US)
ADD $10.45 SHIPPING (US)

PRICED AT **$169.90** (US) SFD with 8050 switch
ADD $10.45 SHIPPING (US)

COMMODORE 8000-9000 SOFTWARE & MISC.

9000 Superpet	$199.95
64K exp for 8032	$110
BPI General Ledger	$25
BPI Accts Payable	$25
BPI Job Cost	$25
BPI Accts Receivable	$25
BPI Inventory	$25
Superscript 8032	$79
Superbase 8096	$79
OZZ Database	$25
Legal Time Acc.	$25
Dow Jones Program	$25
Info Designs 8032	
Accounting System	$50
Superoffice 8096	$109.95
Calc Result 8032	$89
Petcom 1.3 Superpet comm program	$79.95

ORDER NOW WHILE STOCK LASTS!

Send or call your orders to: **Northwest Music Center, Inc. 539 N. Wolf Rd., Wheeling IL 60090. 312-520-2540** For prepaid orders add $25.50 for Superpet, $10.45 SFD 1001, $11.45 B-128. $10.45 4023p, $16.45 9090 and $5.45 64K memory expansion. For software add $3.50 for first and $2.00 for each additional book or program. Canadian shipping charges are double U.S. For C.O.D. orders add $2.20 per box shipped. All orders must be paid in U.S. funds. Include phone numbers with area codes. Do not use P.O. Box, only UPS shippable addresses. A 2 week hold will be imposed on all orders placed with a personal or business check. C.O.D. orders shipped in U.S. only and cash on delivery, no checks. 30 day warranty on all products from NWM, Inc. No manufacturer warranty. NWM reserves the right to limit quantities to stock on hand and adjust prices without notice!

All prices quoted in US dollars.

NORTHWEST MUSIC CENTER INC.

539 N. Wolf Rd. — Wheeling, IL 60090
Hours— (Voice) Phone (312) 520-2540
Mon.-Thur. 12:30-5:00, Sat. 12:00-4:00
(24 Hour Order Recorder)

Super Office
List
~~$149.00~~

SALE PRICE
$49.95*
U.S.

SUPER OFFICE

WITH SUPERBASE 2

ALL THE GREAT FEATURES OF SUPERSCRIPT II AND SUPERBASE INTEGRATED IN ONE FANTASTIC PACKAGE

Store important data then print out personalized custom reports inserting totals and names in the proper places. Do invoices, past due notices and many other things with Super Office. With Super Office you can do anything a small business needs.

Super Office is the ultimate in integrated programs for the "B" series market. Some features include left and right margin settings, tabs, decimal alignment, right justification, underlining, bold print, page numbering and a whole lot more. Super Office is easy to use with simple function key commands and an easy to use manual. There is even a spelling checker to keep you from making embarrassing spelling mistakes. We challenge you to find an easier integrated program or one with more features than Super Office. Includes the ultimate data base at an affordable price. You can do anything that requires data storage with Super Office. Simply set up your screens for easy entry of information (up to 4 screens). Then set up the way you'd like the information to be printed. Super Office does the rest! This program has full calculation capabilities and full data storage manipulation. You do not have to reload either program to access the other. Automated merging through the Super Office application generator.

(must have 256K-1 Meg. upgrade to operate)

* Purchase subject to signed disclaimer.

TAKE A LOOK AT THESE GREAT SUPER OFFICE FEATURES

- ● Help Screens
- ● Bibliographies
- ● Document Chaining
- ● Home Budget
- ● Sales record
- ● Built-in programming language
- ● Inventory
- ● Highlighting
- ● Up to 127 Fields per Record
- ● Calculation capabilities
- ● Formatting
- ● Up to 4 Data Entry Screens per File
- ● Form Letter Printout
- ● Better than dbase II
- ● Mailing list
- ● User definable screen formats
- ● Applications Generator
- ● Up to 15 files per Database
- ● Keyed Access
- ● Report Generator
- ● Macro Capability

Shipping and handling charges $3.95

IF YOU VALUE YOUR TIME, THEN IT'S TIME TO BUY SUPER OFFICE

NWM Rebate Plan

Super Office is shipped with the Superscript II, Superbase I and Super Office Insert

Buy one of our expanded B-256 computers and purchase Super Office for only:

$39.95 with SBI and SSII Manuals

$29.95 without SBI and SSII Manuals

Without B-256 Purchase:

$49.95 with SBI and SSII Manuals

$39.95 without SBI and SSII Manuals

NORTHWEST MUSIC CENTER INC.

539 N. Wolf Rd. — Wheeling, IL 60090
Hours— (Voice) Phone (312) 520-2540
Mon.-Thur. 12:30-5:00, Sat. 12:00-4:00
(24 Hour Order Recorder)

Hacker's Corner

One of a Kind • Surplus • Monthly Special • Closeouts

Limited quantities to stock on hand

PRECISION SOFTWARE

Superbase 1	$9.95
Superscript 2	$19.95
Superbase 2	$39.95
Superscript 3	$39.95

HANDIC SOFTWARE

Calc Result	$79.95
Word Result (special order)	$79.95

B-128 GOODIES

B-128 Programmable Reference Guide was $29.95	$ 5.95
SFD 1001 1 meg drive	$149.95
Superbase The Book	$14.95
Superpet 9000	$199.95
Applied Calc Result	$14.95
The Power of Calc Result	$14.95
8050 rehabs from	$250-$400

Due to popular demand we are now handling the ribbons for the B series printers

MPP-1361 and 8023p ribbons	$5.50
4023p ribbons	$6.00
6400, 8300, Diablo 630 ribbons	$4.95
9 1/2 x 11 continuous form perf (2500 sheets) per box	$24.95

C.A.B.S. ACCOUNTING

General Ledger	$ 9.95
Accounts Receivable	$ 9.95
Accounts Payable	$ 9.95
Order Entry	$ 9.95
Payroll	$ 9.95

SUPER DISK DOC

It's your choice, you can get it now . . .

If you want control over your disks then you need Super Disk Doc!

operates from easy to use duck shoot menus
examines bytes on your disks
interprets in English, hex or ascii
modify and save disk data
recovers previously unreadable files

$24.95

Never before has there been such a powerful and easy to use disk utility like this for Commodore computers!

or . . . you can wait until its too late!

PRINTERS

Gemini 15 x 120 cps	$125.00
Smith Corona DM200 RS232 and parallel (160 cps w/near letter quality mode)	$150.00

ONE OF A KIND ITEMS

EPSON GENEVA PORTABLE. INCLUDES: SOFTWARE, MODEM, ACOUSTIC COUPLER, CABLES, PRINTER, CARRYING CASE. (Floor Model)	$475.00
MJ-10 COLOR MONITOR	$125.00
EVEREX 2400 MODEM	$245.00
MONOCHROME MONITOR	$59.95
9090 7.5 meg HARD DRIVE, REHAB	$495.00

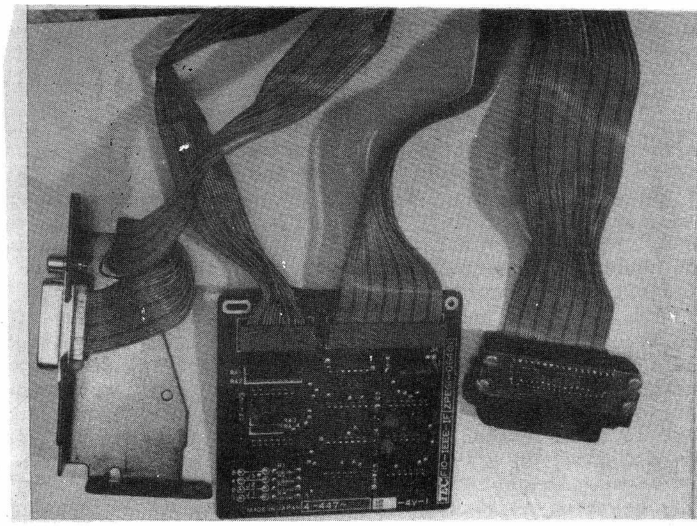
ORDER NOW WHILE STOCK LASTS!

Send or call your orders to: **Northwest Music Center, Inc. 539 N. Wolf Rd., Wheeling IL 60090. 312-520-2540** For prepaid orders add $25.50 for Superpet, $10.45 SFD 1001, $11.45 B-128. $10.45 4023p, $16.45 9090 and $5.45 64K memory expansion. For software add $3.50 for first and $2.00 for each additional book or program. Canadian shipping charges are double U.S. For C.O.D. orders add $2.20 per box shipped. All orders must be paid in U.S. funds. Include phone numbers with area codes. Do not use P.O. Box, only UPS shippable addresses. A 2 week hold will be imposed on all orders placed with a personal or business check. C.O.D. orders shipped in U.S. only and cash on delivery, no checks. 30 day warranty on all products from NWM, Inc. No manufacturer warranty. NWM reserves the right to limit quantities to stock on hand and adjust prices without notice!

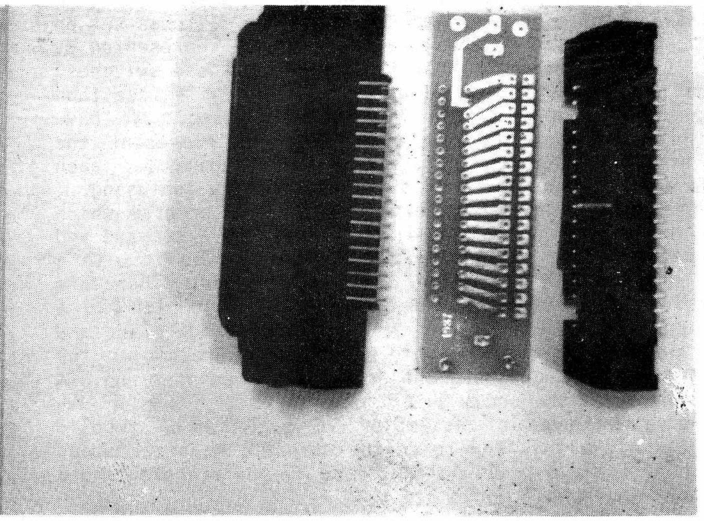
All prices quoted in US dollars.

NORTHWEST MUSIC CENTER INC.

539 N. Wolf Rd. — Wheeling, IL 60090
Hours— (Voice) Phone (312) 520-2540
Mon.-Thur. 12:30-5:00, Sat. 12:00-4:00
(24 Hour Order Recorder)



IEEE to Centronics Adaptor Board
order #11221 $35.00



Connector type changer (comes assembled)
order #11236 $15.00

USE THE 6400 IEEE CONVERTER WITH OTHER PRINTERS

A couple of issues back Warren Kernaghan suggested that the 6400 converter board was a superior method of IEEE to Centronics interfacing. So why not rewire the adaptor board's flat cable to a standard 36 pin Centronics connector? Easy said, not so easy to do even for fairly good technicians. So, CBUG has made up a little circuit board and rounded up the necessary mating connectors to produce a type changer adaptor. Plug the type changer into the Centronics header on the IEEE adaptor, and the other side of the type changer into your Centronics ported printer. Walla, instant Centronics with all the added features Mr. Kernaghan wrote about.

For about a year, CBUG has been offering the IEEE internal converter designed for the CBM 6400 printer. It is a 4" square board with two long flat cables, one of which has a standard IEEE 488 connector, and the other has a 34 pin dual row 1/10" header to connect with the logic board in the 6400. Unfortunately the header with not mate with standard Contronics connectors. The circuit board is double sided plated thru, the connectors prime quality with gold plated contacts.

These adaptors are believed to work with most common printers, though they will not work with my large Daisywriter 2000. They do work with the Star and Cannon printers and many others which now follow the standards. If by chance the adaptor does not work with your printer, we'll refund the full purchase price.

The adaptors have jumper positions to select device numbers other than 4. They are not enclosed so you will want to mount or tape them securely out of the way or install them in a protective box. In some cases, the connector on the printer uses bailing clamps to keep the connectors latched together -- so you may need to use a short Centnronics extension cord compatable with the bailing clamps which are readily available locally or via mail order (see Matos's article).

DISK SALE

5.25"	Each	SAVE!		Each	SAVE!
SSDD	$.57	34%	for IBM AT	$1.70	43%
DSDD	.60	35%	DS4D	1.58	15%
			flippıes	.61	27%
3.5"			8"		
SSDD	1.50	'37%	SSDD	2.60	19%
DSDD	1.70	48%	DSDD	2.80	21%

OPUS QUALITY DISKS!

OPUS disks are without doubt the finest disks made. Mil spec forumulas and exotic materials, promise above spec performance even after 5,000,000 revolutions on each track! Heavy weight jacket and lubricated antistatic liner insure the ultimate in stability. Of course there is a 100% factory warranty backed by CBUG as well.

We are proud of our product. Each disk bears the OPUS corner label; you can be confident you are buying the real article -- no misrepresented no-name seconds. Each disk is inserted in its sleeve; labels + write protect tabs factory sealed in each package. SSDD and DSDD products come 10 in a poly bag; flippies are 25 in a bag. All others are 10 disks in a sealed retail chipboard or plastic package.

This is a special offer by CBUG, Inc. (The Chicago B128 User's Group), one of the world's largest user's groups. Our volume exceeds that of most computer stores and retailers. We pass our good fortune on to you, our members. If you can find OPUS disks in any store, you'll pay several times our price! AND, no one makes a more reliable or longer lasting disk!

------------------------Oct. 15, 1987------------------------

Desc	Qty	Stock#	/Pkg	Extension
5.25"				
SSDD	. .	10017	$ 5.70	$
DSDD	. .	10021	6.00	
DSDD Flippy	. .	10111	15.25	
DSQD	. .	10182	15.80	
For IBM AT	. .	10228	17.00	
3.5"				
SSDD	. .	10074	15.00	
DSDD	. .	10088	17.00	
8"				
SSDD	. .	10509	26.00	
DSDD	. .	10547	28.00	
Pkg 25 Tyvek sleeves	. .	10318	1.00	
TOTAL from other side this form				
TOTAL ORDER				
Ill. Residents add 7% sales tax				
COPY TOTAL TO MAIN ORDER FORM				$

PLEASE ENTER NAME & ZIP BELOW
IN CASE SEPARATED FROM MAIN ORDER

Name ____________________
Zip Code ____________

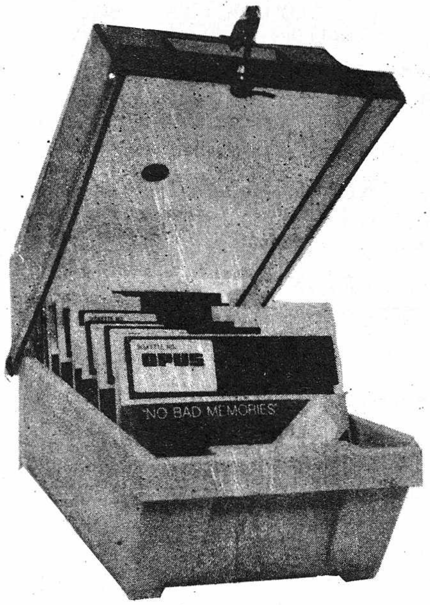
OPUS®

"NO BAD MEMORIES"

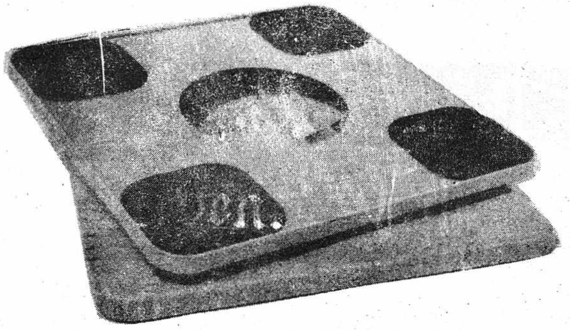
THRU CBUG, PAY THE LEAST!

DATA CASE ORDER FORM

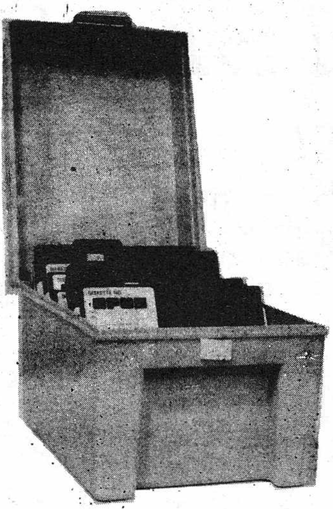
DD120L Holds 120 5¼ Disks
Locking Clear Top, Beige Body
10 Dividers w/ Labels
Order #10960

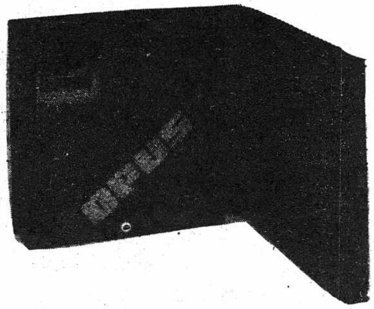


DS100L Holds 100 5¼ Disks
Locking Clear Top, Beige Body
Useable Hinged or Lid Nested Underneath
8 Dividers w/ Labels Order #10975

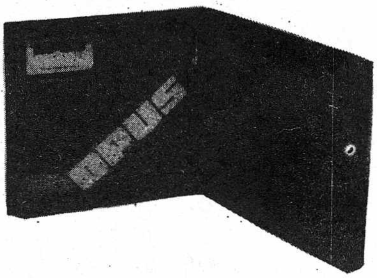


Extra Heavy Duty Monitor Swivel Stand
Holds Monitors up to 12" on Center Feet
Can be used locked in positon or Adjustable
Beige MS1212 Order #10937

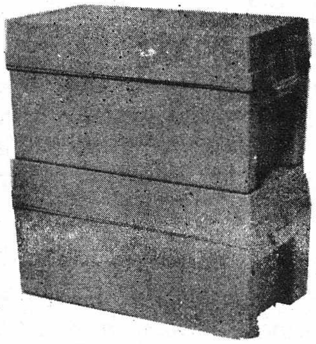




Library Shelf Box, Holds 10 5¼ Disks
Living Hinge Design. Pastel colors
- we'll ship all same in an order
DD10 Order #10989

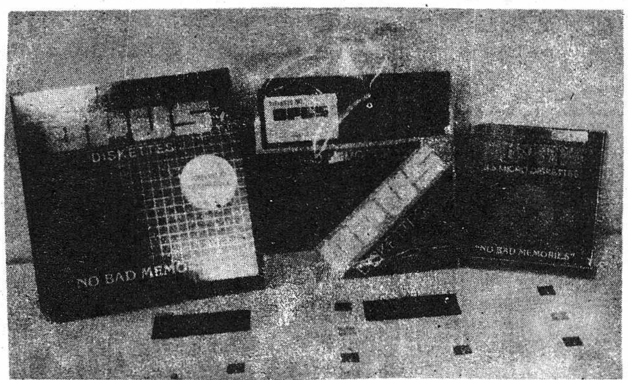


DD5 Mailer/Tote Box, Holds 5 5¼ Disks
Living Hinge Design. Pastel colors
- we'll ship all same in an order
Order #10941



Heavy Duty, Compact, Stackable, Non-Breakable.
One Piece Living Hinge Design. Six Dividers
Factory Installed. Beige body and cover.
D55 Holds 60 5¼ Disks Order #10994
D33 Holds 40 3½ Disks Order #10918

NOTE: Due to shipping and Packaging costs, our data case products are not available outside N. America!



We carry the entire line of OPUS Premium Disks, 3½", 5¼" & 8" Single and Double sided, Single & Double Density; PLUS, quad and High Density AT compatable 5¼ disks.

Item	Description	Quanty	Price	Extension
10960	DD120L locking 5¼ case	____	$16.00	______
10975	DS100L locking 5¼ case	____	12.50	______
10922	DD50L same as 10975; holds 50	____	8.00	______
10994	D55 stackable 5¼ case	____	6.45	______
10918	D33 stackable 3½ case	____	5.00	______
10989	DD10 5¼ library case	____	1.50	______
10641	DD5 5¼ mailer/tote case	____	1.00	______
10937	MS1212 Monitor Stand	____	16.00	______

Name__________________ Zip Code ________ TOTAL ________
Prices and specifications subject to change without notice.

22	"joust"	prg	can your knight beat my knight?
20	"cryptogram"	prg	solve a "cryptic" message.
11	"trendline"	prg	does analysis and forecasting, both "straight line" and "growth cycle".
27	"finance prog"	prg	computes: savings interest, home or car payments, mortgage tables and pension payments.
15	"m/1 code reader"	prg	save machine code as data statements as an adjunct to an assembler.
34	"pricer"	prg	estimates job costs.
11	"plotter"	prg	plots differing variables.
4	"block wall bldg."	prg	provides number of blocks needed, plus cost of construction.
21	"brseat print"	prg	figures seating arrangements for the bridge club.
20	"bar"	prg	prints bar graphs of positive integers.
11	"wallpaper design"	prg	design your very own wallpaper (grandpa's picture included!
5	"concrete pour"	prg	computes amount needed for walks, patios, etc. - with job cost.
4	"footing pour"	prg	figures amount needed - with cost.
25	"draw poker-vegas"	prg	almost like the poker machines in that gambling capital of the world.
23	"encryption"	prg	the "enigma" code of World War II.
13	"dice"	prg	a different kind of "roll the dice".
29	"weather"	prg	amaze your friends - forcast the weather!
16	"logox"	prg	a game - find the right box.
17	"froggie"	prg	a "hoppin" game.
26	"maxit"	prg	played on a chessboard - make the most of every move.
7	"squiggle"	prg	set the parameters and watch the funny line.
5	"lawn!"	prg	instructions for how to mow a lawn
6	"tank!"	prg	instructions on how to destroy the enemy tanks
10	"ram!"	prg	instructions to lead your troops inro enemy land
3	"hawaii!"	prg	cartoon annimation
3	"dromeda!"	prg	an alien arrives on earth (cartoon)
3	"dance!"	prg	an indian rain dance (could use it!)
3	"flight!"	prg	Canadians land on the moon
7	"racer!"	prg	follow the yellow brick road, but don't have a wreck
7	"ambush!"	prg	destroy the forests and isolate the enemy General
27	"!!--notice--!!"	seq	PLEASE READ. It will make you aware of the rules regarding acquisition, submission, usage an dissemination of computer programs!
12	"seq read/print"	prg	included for your convenience in reading the sequential files without benefit of Superscript.
38	"disk contents+"	seq	what you are now reading.
21	"instructions"	prg	full, detailed telling of how to handle this entire rendition!
9	"startup"	prg	included to facilitate getting a "hard copy" listing of all the program files.
92	"african advntre"	prg	rescue poor Dr. Livingston and get some "goodies" too!
29	"swords & sorcery"	prg	a game of "witchcraft".
26	"niche"	prg	an ecological endeavor to preserve and develop the species.
27	"new water"	prg	control the water supply for your town.

847 blocks free.

These are the "8032 emulator" run programs which are also mentioned on the "Master Disk" of this dual disk rendition.

All remain copyrighted, but permission was received to submit them to CBUG for use and enjoyment by CBUG users.

This is a "Support Disk" and its use is fully explained throughout this rendition.

Here is the annotated directory for this disk:

0	"golddisk"	05 2c	disk name
1	" loader"	prg	controls the loading of the
8	" bootscrn .scn"	prg	entry and instruction screens
8	" instscrn .scn"	prg	and loads the 8032 emulator
4	"Start8432.21"	prg	program.
2	"t1"	prg	"
2	"td.alt"	prg	"
2	"t2"	prg	"
2	"td"	prg	"
81	"8432.21"	prg	"
4	"bank F.20 '1024"	prg	"
4	"instructions8432"	prg	"
1	"bm1"	prg	"
1	"bm2"	prg	"
1	"bm3"	prg	"
1	"bm4"	prg	"
1	"bm5"	prg	"
1	"bm6"	prg	"
1	"bm7"	prg	"
1	"bm8"	prg	"
20	"skeet!"	prg	these are the "8032 run"
22	"duel!"	prg	programs - all are fully
54	"drone!"	prg	explained in the "disk
20	"stop!"	prg	contents+" program on the
17	"voz!"	prg	"Master Disk".
26	"hawaii!"	prg	"
23	"tank!"	prg	"
23	"ram!"	prg	"
15	"lawn!"	prg	"
15	"blasto!"	prg	"
28	"dromeda!"	prg	"
23	"dance!"	prg	"
14	"ambush!"	prg	"
19	"racer!"	prg	"
26	"flight!"	prg	"
10	"disk contents+"	seq	what you are now reading.
27	"!!--NOTICE--!!	seq	PLEASE READ and be guided by its content.
12	"seq read/print"	prg	included to allow reading of the above two sequential files without resorting to Superscript.

1532 blocks free.

RESTAURANT ACCOUNTING CBUG #92 NEW RELEASE 2808 #13156

By: Lowell Breunig

Please see the complete article this issue describing these programs in detail.

TSUM and MOLG

TSUM and MLOG are companion programs written for use in a Restaurant, however they can be used by other business or private use. TSUM is an internal invoice, Sales and theft control program. It's purpose is to supply operational control data on daily basis and retain that data for future use as needed. Data input from source documents, is fast, with easy corrections performed during input.

TSUM, is impossible to describe in a paragraph. Two of its best features are the ease and speed of input with instant price verification. The simplicity of number series tracking, such as invoices, will blow your socks off. Anyone who needs to control Guest-Checks, Invoices, Purchase Order forms or any other need to track items in series should see this routine.

MLOG and TSUM are companion programs, however MLOG can be used as a stand alone program. The purpose of MLOG is to set up a list of items within 14 categories with a unique code number for each item. It can be used as an inventory list, phone book and/or address book.

If you are learning to program these programs have a number of routines that are easy to pull out as a block and play with. All the major variables are listed in rem statements at the top of the program and all routines separated by rem statements.

```
1 "G/L Data Disk    " 01 2c   3   "luncJul83"      seq   1   "miscJul83"          seq   18   "newsletter intro" seq
134  "tsum"             prg   2   "dinnJul83"      seq   1   "beerJul83"          seq   3    "intro. norman"    seq
35   "mlog"             prg   1   "salaJul83"      seq   1   "giftJul83"          seq   26   "P/R Op-instru"    seq
2    "dessJul83"        seq   1   "soupJul83"      seq   2   "guest checks out"   seq   6    "pos."             prg
4    "breaJul83"        seq   3   "beveJul83"      seq   5   "g/c loader"         prg   1701 blocks free.
1    "add-Jul83"        seq   2   "sideJul83"      seq   10  "mlog Instruction"   seq
2    "burgJul83"        seq   3   "lighJul83"      seq   85  "tsum instruc."      seq
```

B-128 SERVICE INFORMATION CBUG #93 NEW RELEASE 2807 #13160

By: John Plosila

Mr. Plosila has spent a lifetime writing techical instructions for one of the major electronic/communicat manufacturers. Over the last two years he has been studying and inquiring of the B-128's construction and ope principles. The fruit of these efforts is on this disk. It is an important acquisition for both the technici programmers seeking a better understanding of how our machine operates. There are numerous references in John's ai es to the B-128 Service Manual. It is identified as Commodore Business Machines "Service Manual, B Model Computer, Jai 1985 PN-314010-06. It may be purchased direct from Commodore for $25.00 plus $5.00 shipping & handling (US). Write directly to CBM for the service manual: CBM, Parts Dept., 1200 Wilson Dr., West Chester, PA 19380. As of September 1988 there was more than sufficient supply on hand to meet the needs of CBUG members. We strongly suggest you obtain the service manual along with Mr. Plosila's articles on this disk.

```
1 "b128              " aa 2c   7   "disk/out"   seq   44  "modem"      seq   41   "irq1"    seq
11   "disassembly"       seq   4   "keyboard"   seq   8   "1200"       seq   128  "irq2"    seq
42   "disassembly/serv"  seq   3   "timing"     seq   6   "db"         seq   95   "irq3"    seq
19   "mpu"               seq   4   "sound"      seq   39  "aciatest"   seq   126  "irq4"    seq
11   "ram"               seq   3   "pwr"        seq   17  "diskop"     seq   41   "irq5"    seq
8    "rom"               seq   12  "acia"       seq   85  "6509"       seq   1182 blocks free.
7    "video"             seq   17  "rs232"      seq   92  "irq"        seq
```

FAST BUS CBUG #84 2800 #13033

By: Dennis Jarvis & Jim Springer

This disk was discussed at length in an article in the last issue of The CBUG ESCAPE, Vol 10, Winter 1988. Additiionally there is an article this issue suggesting the applications and benefits of the program. Fast Buss is the software portion of a package which allows for the connection of any Commodore Serial Device to the B128. In addition to the software, you must have a 24K RAM Cartridge such as the one sold by Anderson Engineering, and also the Fast Bus adaptor also sold by Anderson Engineering. See the ad in this issue for those items.

There are a number of other programs and data files of considerable value on this disk as well which can be used with or without the Fast Bus products.

```
1 "b128 fast bus V2" fb 2c     70   "disk copier"     prg    21   "&universal trans" seq    127  "jump table 2"     seq
```

```
"loader"            prg   62   "make 2 sided"      prg   17   "&make 2 sided"     seq   1    "<-----misc----->" prg
"read me first!"    prg   44   "universal trans"   prg   12   "&convert"          seq   10   "what's next"        seq
"read me first"     seq   1    "convert"           prg   1    '<--technical--->"  prg   10   "bam copy 4040"      prg
"b-128 fast bus.2"  prg   1    "<-instructions->"  prg   90   "memory map"        seq   35   "fast bus article"   seq
"<basic programs>"  prg   15   "&disk copier"      seq   56   "jump table 1"      seq   3    "--!!NOTICE!!--"     seq
                                                                                          1439 blocks free.
```

CPM/86 2.003 CBUG #PR 18 NEW RELEASE 2809 #13103

By: John Wright

This disk is more a demonstration disk than anything else. I have found an 8080/Z80 emulator that will run on the B-128. Most of the files on this disk are 8080 programs. The 8080 programs are the ones with the extension of COM. To use the emulator simply type the following:

<<NOTE: This disk requires the 8080 co-processor board to be installed in your B-128.>>

A:Z80 Filename (command tail)

Where the command tail is the instruction the program requires. That is if you want to run the SQ.COM program the "Command Tail" would be the File that you wanted to Squeeze. For example:

A>Z80 SQ FLN.TYP Will envoke the Z-80 emulator, run the program SQ and Squeeze the file "FLN.TYP".

The catalog file gives the usual listing of files and a short description of what they are/do. Again as with my other disks, I have included a help file. To obtain assistance you simply type "help+ disk and you will receive a menu . Select the document you wish to read, and it will be printed on your screen. Unlike my other disks, this one does not contain the "DISK.LBR" file. On my other disks this file was used with the HELP+ pgm to provide the help files. I had some "checksum" problems with this disk so just included the document files in the DISK.HLP file. It is a little slower, but at least the utility is still available in the "DISK.HLP" file. Sorry about the inconvience.

I have also included the C128 CPM version of Kermit. Kermit is a telecommunication transfer protocol program that we use on the DDN. This particular one emulates the VT-52 terminal. It doesn't run because I haven't figured out how to address the RS232 port on the B from within CPM. If anyone gets it to go, I'd appreciate knowing how you did it. This is a Z-80 program, (that's what the C-128 uses) so you will need the Z-80 emulator to run it.

There are some CPM-86 programs as well. The SCRUB command will clean up a Wordstar file and make it readible using the "TYPE" command. I have used it on this file.

Most of the programs on this disk are duplicates of previously provided programs. Again this is more a demo disk than anything else. I just thought that you might be interested in a 8080 emulator. This could be especially usefull if you want to get some programs from a local BBOARD.

Have fun with this disk. If you need help, let me know. If you want to help PLEASE let Norm and me know.

Lt Col John A. Wright
818 Juniper Dr.
Papillion, Ne.
(402)-339-5728

CONTINUED ON NEXT PAGE

CATALOG OF CP/M-86 UTILITY DISK 003

PROVIDED BY:

LT COL JOHN A. WRIGHT
818 JUNIPER DR., PAPILLION, NE.
68046
Tele: 402-339-5728

```
*****************************************************************************
********          NOTICE!!!:  All programs copyrited, and not      *********
********          Releasable outside CBUG, nor releaseable for     *********
********          for commercial purposes!!!                       *********
*****************************************************************************
```

003.01	INSTRUCT	.HLP	4K	97 CE	Disk introduction
003.02	DISK	.HLP	34K	03 43	Help file used with Help+
003.03	HELP+	.CMD	6K	65 1E	Help program
003.04	BUFFLIB	.REL	6K	DC DA	? Left for the curious
003.05	BUFFLIB	.SZR	4K	8A 4B	/
003.06	C128KER	.COM	15K	42 F9	Kermit for the C-128
003.07	C12BC	.MAC	1K	DC 9C	/
003.08	CCITCRC	.COM	2K	6A D2	Similiar to CRC
003.09	CPYRITE	.MAC	1K	C9 31	Copyrite notice
003.10	CRC	.COM	3K	B2 07	CRC check routine
003.11	CRN	.MZC	18K	E4 C1	? Part of a Crunch package
003.12	CRN	.REL	2K	55 DB	/
003.13	CRN	.SLR	2K	78 52	/
003.14	CRUNCH	.COM	5K	1F DE	Crunch files (similiar to SQ)
003.15	CRUNCH	.MZC	11K	64 45	/
003.16	ERAQ	.COM	1K	3E A1	Selective erase
003.17	ERASE	.COM	7K	E7 CF	Erase program
003.18	FIND	.COM	2K	BA F8	Find a string in a ASCII file
003.19	FYNDE	.COM	3K	FF BA	Similiar to FIND
003.20	IDIV	.MZC	1K	BF B9	?
003.21	IMUL	.MZC	1K	46 46	?
003.22	INDEX	.COM	11K	41 14	Build and index
003.23	LU	.COM	20K	77 39	8080 Library Utility
003.24	MLOAD24	.COM	3K	D8 AA	8080 File linker
003.25	NEATC	.COM	2K	9A C5	Cleans up source files
003.26	NEATL	.COM	2K	3B 2B	/
003.27	NEATU	.COM	2K	37 C5	/ (Read documentation)
003.28	SCRUB	.CMD	14K	1E 79	Cleans up Wordstar files
003.29	SHOW	.COM	3K	F6 28	Replacement for TYPE
003.30	SQ	.COM	17K	66 FE	Squeeze a file
003.31	TYPEL	.COM	4K	0D E9	Replacement for TYPE
003.32	UNC	.MZC	12K	B0 AE	? Uncrunch files
003.33	UNC	.REL	2K	F4 D3	/
003.34	UNC	.SLR	2K	5E 3A	/
003.35	UNCR	.COM	4K	D8 FA	Uncrunch files
003.36	USQ	.COM	2K	56 21	Unsqueeze squeezed files
003.37	WHATSNEW	.CMD	3K	80 85	What files have been added to disk
003.38	Z80	.CMD	5K	D1 50	8080 emulator (supposed to be z80)
003.39	Z80	.LBR	31K	F0 74	Z80 Library of files as downloaded
003.40	!!--NOTI	.CE-	1K	90 27	Public Notice

Software Tools RCPM - MISC Volume Number - 003, 41 Files cataloged.

ESCAPE PRINT FILES Vol 11 Spring 1988   CBUG #94   NEW RELEASE   #13175

The complete print files for the Spring 1988 issue of The CBUG ESCAPE. No directory available at the time this issue's library listings are going to the printer.

LEGAL CORNER

By: F. Petersen, Esq.

Re: Copyright Matters

Some questions have been raised concerning the matter of copyrighted material insofar as individual computer programs are concerned.

This is not meant to be an "all inclusive" copyright treatise and only those matters which could affect individual programs will be addressed.

Note: Much of the material herein has been gleaned from an article by Jay Stuller, titled "Your Guide to Copyright" which appeared in the June 1988 issue of Writer's Digest.

Although not specific as to computer programing and the resultant computer programs, it appears the following data is applicable for our purposes.

First - a little history.

Copyright law provides basic, automatic protection for writers whether or not a manuscript (e.g. computer program) is registered with the Copyright Office or even published. Under the Copyright Act of 1976, which took effect in 1978, an "original work of authorship" has copyright protection from the moment the work is in fixed form. That is, as soon as one has an article, short story, book or program, etc. on paper, on a computer disk or even spoken into a tape recorder, it is protected by copyright law.

Next - what can be copyrighted?

This protection exists for broad categories of works including computer programming.

Copyright law generally gives the owner of the copyrighted material the exclusive right to reproduce and to authorize others to reproduce the work, to prepare derivative works based on the copyrighted material and to display the work publicly. Note the word "generally" for copyright law includes a number of limitations and exemptions.

Several categories of material are not eligible for copyright protection, such as titles and short slogans; works consisting entirely of information taken from common sources and public documents, such as standard calendars, lists and tables; and speeches and performances that have not been fixed on paper or recorded. Also, work in the "public domain" (material whose copyright has lapsed or that was never copyrighted), material that lacks sufficient originality and basic themes and plots cannot be protected by copyright.
Neither can ideas be copyrighted.

Since the idea isn't copyrighted, what is? The form of the idea, how it has been expressed, how the words (or program) has been strung together.

Also to be considered is the so-called "fair use doctrine".

Fair use, under copyright law, allows people to use copyrighted materials in limited ways: brief quotes in other manuscripts (computer programs), excerpts in book reviews, small numbers of copies distributed in classrooms.

To gain copyright protection the work must be "original", which means you cannot copy from or infringe on another writer's copyrighted work. This does not mean that something you write can't occasionally be similar to what someone else creates and vice versa - in terms of the topic, tone and even the substance.

You can, of course, use the information and facts found in copyrighted material without infringing on it. Moreover, "fair use" allows one to use parts of the whole. This is, after all, what a lot of research is all about. It is best; however, to always put things in your own words, to cite the source and to be very judicious in the use of other people's material.

The life of a copyright.

Under current law, copyright protection lasts for the life of the author plus 50 years.

Works created anonymously or pseudonymously have 100 years of protection after creation or 75 after publication, whichever is shorter. Works created "for hire" also come under the 100/75 rule. When a copyright expires the material enters the "public domain".

Prior to the Copyright Act of 1976, material published without a copyright notice, or an improper notice, could fall into the "public domain". Today such a problem, even if the material is published without a copyright, can be remedied if a reasonable effort to place a notice on all undistributed copies and to register the material with the Copyright Office within five years of publication is made.

What's in a letter?

A "c" inside a circle indicates copyright ownership while an "r" inside a circle indicates that a word or phrase is a registered trademark.

How is Copyright material registered?

Complete a form provided by the Copyright Office, include the $10.00 fee and one copy of the material if it is unpublished or two copies if it is published. Send all to the Register of Copyrights, Library of Congress, Washington, D. C. 29559.

Also, large numbers of materials can be registered simultaneously for the same $10.00. Or you can spend $120.00 for fast Copyright Office service.

Finally.

Why bother to register the material if it is automatically copyrighted from the moment it is in fixed form?

Beats me. However, Victor Martin of the Copyright Office recommends registration mainly for legal reasons. "If your work is infringed upon, registration established a public record for your claim and prima facie evidence that a court will accept", he says.

Nuff for now!

◆◆◆◆◆◆◆◆

Comments and Suggestions for Protecting Donors of Software

By: Mathew Goldstein

<<This article is abridged. Full text on the Print File Disk for this issue.>>

I would like to respond to some questions which have been raised about the protection of software released thru the CBUG library. The primary objection, as I understand it, has to do with legal problems and inconveniences for the programmer resulting from changes made to the program. One way for contributors to protect themselves when they send no charge programs out for publication, with a minimal investment of their energy and time, is to mail a printed copy of the program to themselves in a sealed envelope (a notary public can witness the sealing of the envelope and/or the enveloped can be mailed with a return receipt request for added security) <<or better yet, the time proven registered letter to yourself containing the evidence -- to remain sealed till needed and opened in court>>. Some CBUG programmers could purchase a compiler and compile their BASIC programs for an added measure of security. Everyone benefits from faster, more compact programs. Compiled code is much more difficult to modify. It can only help to pay the $10 fee to register your program with the copyright office, particularly if you are charging for your program. To apply for a copyright call 202-287-9100 and ask for form TX and circular 61.

Contributors are entitled to seek a small profit from their contributions, and some have, others have not. I intend to mix public domain contributions with contributions which I will charge for. We need both types of contributors, I think, the former for all members to acquire programs that are affordable and to allow other programmers to learn and to build upon each others work, the latter to provide incentive for the long hours and effort required to produce more powerfull programs which some of our members will need. Appearantly, few people give freeware donations, I think I received one freeware donation to date. To keep things in perspective, bear in mind that no matter how you slice it there will always be ambiguities and risks associated with publishing software, whether it be public domain or copyright.

BASIC SYNTAX CHECKER

With Non-existant Branch and Unutilized Code Analysis

<<A very important program for <u>ALL</u> programmers!!>>

By: Mathew Goldstein

Some of the public domain programs on some of the CBUG disks do not work. In some cases this is because the programs, as written, do not correctly perform the desired function. If you are able to recognize and correct logical errors in public domain programs you can, of course, share the correction with everyone through CBUG.

In many cases programs do not work because of illegal command and statement formats. Sometimes this happens because the programs were written for Commodore computers such as the C64 or PET which are not fully compatable with our 'B' Series computers. Some of these programs may run in the 8032 virtual machine environment. You can use the 'list/conv' program on CBUG #46 or the 'list all' program on CBUG #17 to help in identifying which Commodore computer the program was written to run on. Other times the person who wrote the program mistyped a line or inadvertantly used an incorrect command format. I have written a program which should identify such syntax errors in Commodore BASIC 4.0 programs.

<<the up arrow symbol is not available on our printer -- it is represented by "²". Again, the disk version is much easier to read.>> The syntax checker displays or prints an error message preceded by the linenumber of the code in error and followed by a summary of the command line up to the place where the error was found. The error messages use a few abbreviations including 'stfn' and 'nmfn' for string function and numerical function respectively. An example of a string function is chr$(and an example of a numerical function is' asc(. The abbreviation for variable is 'v.'. All numerical operators (+, -, *, /, ², and, or) are represented by '+'. Paired logical comparison operators (boolean operators) are represented by the first symbol. Thus '<=' is represented in the error messages as '<'. Syntax errors which occur after the 'then' in an 'if ... then ...' statement will omit the 'if ... then' portion of the statement in error messages whenever 'then' is followed by a keyword token (such as PRINT). The error messaging has undergone improvement after the results below were generated. The error message for 'insfile' on CBUG #5, for example, is now:

```
7110 missing =: FOR v.
15018 missing numeric operator: PRINT "anjualpremium";TAB(2
```

I ran the 'BASIC 4.0 syntax' program on the following CBUG disks: 1, 2, 3, 4, 5, 6, 11a, 17, m20, 27, 28, 29, 32, 46, 48, 54, 55, 57 and 66. Below is a listing of the syntax errors found, excluding those programs written for different Commodore computers which contained many errors.

```
CBUG #1
"name sort"         unclosed paranthesis line# 1118
"disk directory"    mispelling or illegal expression line# 1
             missing numeric operator: If v.=num. line# 160
                    missing (: v.$=stfn line# 2940
"dfreport"          Go without TO line# 782
                    Go without TO line# 786

CBUG #2
"change dev#"     illegal printlist: PRINT string; line# 220

CBUG #3
" star trek 2"       missing numeric operator:
                            IF v.+v.=num.THEN v.=FNname(v.)+
                            nmfn(²)+num.+num. line# 1830
" star trek 1"       same as star trek 2
" eliza"             missing $: v.$=stfn(v. line# 410
" calculator"        error: line numbers must increase
                     sequentially in BASIC programs, 257
                     follows 260

CBUG #4
"body weight"      illegal printlist: PRINT string line# 195
"checkers80"        missing =: v. line# 5840
"banner"            missing terminator: CLOSE num. line# 940
"big letter ads"    missing numeric expression:
                       IF v.$=stringTHEN v.$=stfn(line# 530
CBUG #5
"discbal"        mispelling or illegal expression line# 503
"deprsf"         GO without TO line# 190
                 missing =: v. line# 3310
                 illegal operator: PRINT line# 10010
"timeclck"       missing =: v. line# 10010
"upszone"        missing =: v. line# 340
"autoexp"        illegal printlist:
                         PRINT TAB(num.);v.$line# 16005
"invoice"        missing =: v.$ line# 1852
                 missing =: v. line# 1852
                 mispelling or illegal expression line# 1853
                 mispelling or illegal expression line# 1854
                 missing string operator: IF v.$(string
                     line# 1855
                 illegal printlist: PRINT#num.,TAB(num.
                     +v.)v.$; line# 1857
"insfile"        illegal variable after FOR line# 7110
                 missing numeric operator: PRINT string;
                     TAB(num. line# 15018
"invent2"        missing =: v. line# 2301
                 missing =: v. line# 3730
                 missing numerical expression: IF v.<num.+
                     line# 5072
"busbud"        mispelling or illegal expression line# 13631
                missing =: v. line# 13631
                missing =: v. line# 13631
                missing =: IF v.<num.+v.<num.+v.<num.THEN v.
                     line# 15540
"payroll2"      missing =: v. line# 10020
```

<<The next 90 lines of listing have been deleted from the printed text.. Full text is available on the PRINT FILE disk for this issue of The CBUG ESCAPE. The text also appears on the SYNTAX program disk listed in this issue's library section.>>

As you can see above, the program has undergone extensive testing and has demonstrated that it can handle the undocumented nuances of BASIC 4.0. For beginning programmers it serves as a BASIC programming tutorial and experts will find it a valuable aid when writting lengthy programs. The program will optionally perform an unutilized code search. Because 'BASIC 4.0 syntax' is compiled it will run fast, syntax checking even large programs within minutes.

SS2 TABLES:
GRAPHICS CHARACTERS AND SPECIAL EFFECTS ON 4023 PRINTER

By: Mathew Goldstein

To be able to use all of the graphics characters available when printing Superscript II documents bload into bank1 starting at location 57344 the final version of the "ss2 tables" program developed by Neil Cumfer. Then print this document for reference

<<Due to our inability to reproduce most of the control characters as seen on the screen, the tables below stand as a brief example of the materials on the disk. You must read the article on screen from disk to get full benefit.>> <<ABRIDGED: We have attempted to reproduce two examples of the 116 lines of data compiled by Mr. Goldstein. It is necessary to read this file from disk. It is on both the Print File disk, and Mr. Goldsteins library contributions this issue.>>

The paranthesis around (esc >) means that graphics mode is optional for that sequence

The additional polka-box graphic above is shown in the gray manual, page 115, as screen-display poke-code 94 (set 2)

It easier to remember all of this if you start by promising not to use shortcuts or exceptions other than those required as noted below. Note the uses of escape 1, escape 3, escape 7 and escape 9.

memory hints for using superscript II graphics

1. GENERAL RULES:

a) Remember that for all sequences that begin with esc >, the esc > prefix sets the upper/graphics mode and once set remains in effect untill the end of the line. Therefore it need not be repeated for each upper/graphics mode character that occurs in sequence

b) The esc > sequences take up one additional space on the screen but not on the printer. The same applies for the other esc sequences used for special effects.

c) Although many of the graphics are accessible in lower/upper mode (without being preceeded by esc >) it is easier to use upper/graphics mode at first to avoid encountering difficulties with the exceptions. Concentrate instead on the exceptions involving the v, n, and l keys, the odd/even numeric keys on the main keyboard, the keypad /, the four additional graphics symbols not displayed on the keyboard, the British-pound, left-arrow, up-arrow and pi characters, and the brackets

2. ALPHABETIC KEY GRAPHICS: all of the graphics on the undersides of the alphabetic keys on the main keyboard are accessible with and esc > shift sequence followed by the key with the desired graphic with only three exceptions:

a) The middle horizontal line graphic found under V is accessed by esc-1

b) The middle vertical line graphic found under the N is accessed by esc-3

c) The graphic under the L uses esc > shift V

the L, N, and V graphics are mislabeled on the keyboard, the graphics for the N and L keys are in fact the same as

SPECIAL EFFECTS <ESC> SEQUENCES

effect on 4123 printer	keystroke sequence	screen graphic	
reverse on	ESC <	reverse R	BN
reverse off	ESC >	reverse r	BN
enhance on	ESC + (left bracket)	reverse W	BN
enhance off	ESC I (right bracket)	reverse N	BN
carriage return without linefeed	ESC ′	reverse left-arrow	BN
upper/graphics mode (temporary)	ESC >	reverse u	BN
lower/upper mode	ESC ;	reverse l	BN

GRAPHICS CHARACTER SEQUENCES

actual char.	ascii	ss2 sequence of char.	actual char. screen only	notes:
B+	91	escape > left bracket	+	bracket is a graphic
[		(or control keypad 3)		
£	92	escape $	░	
B I	93	escape > right bracket	I	bracket is a graphic
B※	94	escape > up arrow	※	up arrow is a graphic
↑		(or pi)		w you see is not w y get
	95	control left-arrow	left arrow	(will not display)
▮	161	(esc >) control 1	▮	
▪	162	(esc >) escape 2	_	esc not ctrl for even no.'s
‾	163	(esc >) control 3	‾	numeric key graphics
Z	164	(esc >) escape 4 or control 4	_	except for key 6 are
I	165	(esc >) control 5 or escape 5	I	accessible in lower/
	166	(esc >) control pi		upper mode also
BN	167	escape > escape 6	graph	must use esc > !
⁂	168	(esc >) control ′ (or control ")	⁂	
text	169	control = (or control +)		graph not as under key!
B graph	169	escape > control = (or +)	graph	(as under key)
I	170	(esc >) control keypad −	I	
⊦	171	(esc >) control keypad +	⊦	
L	172	(esc >) escape 0		esc not ctrl for even no.'s

the graphics for the left caret (less than symbol) and left bracket.

3. NUMERIC KEY GRAPHICS: the numeric key graphics on the main keyboard are accessed using an esc > control sequence for odd numbers and an esc > esc sequence for even numbers.

4. NUMERIC KEYPAD GRAPHICS: the graphics on the numeric keypad are all accessible with an esc > control sequence with one exception: the graphic under the / is only accessible using an escape 9 sequence.

5 ADDITIONAL GRAPHICS: there are four graphics which are not shown on the undersides of the keys that generate them. These four graphics are generated by the right-bracket, up-arrow, +/= key, Biritsh-pound/left-arrow key.
a) A check-mark graphic is generated using control right-bracket in lower/upper case mode only
b) A polka-boxed graphic is generated using the up-arrow (shift 6) in lower/upper mode only
c) A striped graphic is accessible with a control +/= in lower/upper mode only
d) A striped graphic is accessed using esc-7.

6. SPECIAL CHARACTERS: there are also four special characters; left-arrow, British-pound, up-arrow, and pi which require special sequences to access.
e) The left-arrow graphic is generated using a control left-arrow sequence
f) The British-pound graphic is accessed using esc-$
g) The up-arrow graphic is accessed by esc > control 6 or esc > control up-arrow (shift 6)
h) The pi graphic is accessed using esc > ^{2}. It is not accessible in lower/upper mode

7 BRACKETS: the brackets are treated as graphics symbols. Use esc > left-bracket and esc > right-bracket sequences

◆◆◆◆◆◆◆◆◆

CHANGING IEEE DRIVE UNIT ADDRESSES VIA AMPERSAND FILES

By: Anthony Goceliak

(8050 dos 2.5, 8050 dos 2.7, sfd-1001, 8250, 9060, or 9090 all respond to these ampersand files correctly).

First off, let me assure you that you don't NEED to use this approach to alter the unit address of your drives, but I have sure found it to be useful.

By cutting one or more jumpers as described in the CBM drive manuals, the power-up unit address may be changed, but only within the range of 8 to 15 for 80xx/90xx units. Maybe you feel unqualified to perform surgery between all those itty-bitty p. c. board traces, or perhaps you don't want to forever commit your drive to a life as unit #14, or perhaps you really need to set the unit address higher than 15 or lower than 8. Following the relatively simple instructions in the drive manuals can result in a soft change within the full available IEEE address range, but this too has a serious drawback. How do you configure your system when two or more units power-up as the same unit, (usually 8)? It is a royal pain to have to turn power switches on one at a time and type in print#15,"m-w"chr$........ several times running.

This approach can configure any number of drives which power-up originally as the same number all at once, and as long as the files remain available to the drives, can easily provide 'shift-run' system configuration using no operator intervention.

If all your drive units are as they came from the factory, they are set as unit #8. By copying only one address changing ampersand file to a disk (which may have any number of additional files, programs, or whatever in addition), and inserting the disk so set up in drive #0 of each unit whose unit address is to be changed, all drives can be configured at once by typing the following:

open15,8,15,"&drv adr #*":close15 (and of course pressing return)

Each drive will spin and select the ampersand file it first encounters in the directory which pattern matches the above, or in other words it's address gets changed to whatever number the file on disk says! The command may equally easily be given from within a program, enabling you to set up a drive to whatever address you desire, without regard to how the user's system was originally set up. The unit address change will remain until the drive is reset, either by another address change, by sending a 'uj' command properly to the drive, or by powering down.

Alternately, by typing the full filename without pattern matching, you can re-configure one drive at a time, since the remaining drives merely turn their error leds red (file not found). Using this method does not require copying the ampersand files and offers complete flexibility in choosing which drive is unit whatever today.

e.g. :
open15,8,15,"&drv adr #19":close15 (and return)

For those who may be interested, configuring the 80xx to a unit address below 4 is possible, but your CBM computer will no longer find the drive unit, since your computer believes that numbers 0 through 3 are not IEEE devices!

Mr. Anthony J. Goceliak
32 Cottage Street
Jersey City N. J. 07306

◆◆◆◆◆◆◆◆◆

CREATING, MANIPULATING, AND USING DISK FILES WITHOUT NAMES!

By: Anthony Goceliak

Once again, DOS has proved more flexible than your b-computer when the subject of filenames is approached. Although your b shrieks "ERROR!" when you attempt to save or open a file without a name, your good ol' 80xx will happily do any legal file manipulation for a file whose name is nothing.

What we have to do is convince the b that there is something there, even when there isn't, and both halves of the dynamic duo become happy, and the file can be used. The proceedure is simple.

From basic, assign the variable f$ the value of chr$(160), which is the dos inverted space end of filename marker.

```
f$=chr$(160)
```

Next, lets create a sequential file, (although any type is ok)

```
dopen#1,(f$),w
```

Drive #0 of unit eight whirs, and presuming there is a disk, the file opens, and presto! we're home!

```
print#1,"data"
print#1,"more data"
print#1,"etc....."
```

Write what you wish, the file is open. Now to close:

```
dclose#1
```

Try this on for size, type directory d0 and somewhere in there will be a file that looks like this:

```
1   ""                 seq
```

The no-name file.

You can read back the contents of the file by the following elementary demo:

```
10 f$=chr$(160)
20 dopen#1,(f$)
30 input#1,a$:print a$
40 input#1,a$:print a$
50 input#1,a$:print a$
60 dclose#1
```

Type it in and run it. When you are finished, try this:

```
f$=chr$(160):scratch(f$)
```

If you have scratched the file, next try this:

```
dsave(f$)
```

And last but not least, power down your b and then type the following:

```
f$=chr$(160):dload(f$)
```

It all works. Rel files and Usr files may also be created without name, and no-name files can be handled perfectly by all the DOS commands which I have tried. The possibilities are there, the uses are limited only by your mind.

Mr. Anthony J. Goceliak
32 Cottage Street
Jersey City N. J. 07306

SORTING FILES THE SNEAKY WAY

By: Anthony Goceliak

Once again I hope that you have read my previous Escape article concerning file renaming, since it forms the basis for this quicky.

To sort all files contributed by programmer 'a', or whatever catagory 'a' means in YOUR personal filenaming convention:

```
directory d0,"?????????????"+chr$(160)+"a*"
```

or to copy from d0 to d1 each of these files:

```
copy d0,"?????????????"+chr$(160)+"a*" tod1,"*"
```

Remember that dos 2.7 at least (2.5 too?) bombs the drive if the pattern existed on d1 BEFORE the copy command was given, so using the command in this fashion always allows a whole diskful of category 'a' files to be added to an ordinary disk, since no 'usual syntax' files can pattern match the shifted space in the fourteenth place followed by an 'a'.

Mr. Anthony J. Goceliak
32 Cottage Street
Jersey City N. J. 07306

USE OF DUPLICATE FILENAMES ON COMMODORE DISK-DRIVES

By: Anthony Goceliak

Prologue

This is a re-creation of my original article which was the victim of my b's power supply, which failed EXACTLY as I wrote the tag-line and was about to file it! Unfortunately, the original article had much greater literary interest, but I hope this one has equal information. I want to thank Norman for locating the critical part responsible for the failure, which turns out to be a very hard to find critter. Due to our user-group the part was only a phone call away. Something worth considering at renewal time perhaps.

LOGUE

As we all know, CBM has prevented us from using duplicate filenames, right? ....right?? .....well, not exactly.

If you read my article on file re-naming in the Summer 1987 Escape, you will understand the significance of the apparently unorthodox filenaming conventions listed below, but they are indeed legal, and can result in a pair of (or a hundred!) files each 'legally' named in an identical manner, and each Individually and Correctly addressed by your drive!

In a bit plainer language, I am gonna show you how to create two basic program files, both referred to by the name "test pgm", and demonstrate how to dsave both of them to the same disk. Furthermore, if you put the disk away and shut off your b for a month, when you power up again, you will be able to correctly dload either one, at YOUR selection, not the b's. And last but certainly not least, show how to scratch just one, or both or (in the case of the hundred files) all of 'em!

Right about now when I read one of these 'gee whiz' articles, I start to ask "so what? what's it going to do FOR ME!". Fair enough.

-Have you ever written a basic program? (or m/l, or just about anything?)
-Has it ever taken you more than one try to get it right?

If you answered either of these two questions with a 'no', don't bother to read further, you're either too smart or too .... to need any help that this article can provide. (By the way, the .... in the prior sentence was not edited out by Norman, but I can think of five different words to fit there, only three of which should be considered negatively).

AHA! the doubting Thomas from the third row says 'What's the matter with good old dsave"@test pgm"?' Absolutely nothing I reply, JUST SO LONG AS YOU NEVER MAKE A MISTAKE, and each and every 'newer' version is one step closer to the final product!

However, for those of us who try out this way and that, just to see if performance is better, or faster, this approach saves every copy of your program as you develop it, modify it, or abandon it, until you are satisfied with the final version. Nothing is ever lost to a promising attack on a problem which just frankly got bogged down by unforseen complications, (or just frankly worked not-at-all!).

THE DEMO

First let's set up our test program. For once I don't care if you're running 'Keytrix', since this relies on finesse rather than re-programming of the operating system. However, at least type 'new' (and return) from basic to clear out any old programs that might still be lurking in memory.

Next type the following lines:

```
10 print"this is version number one."
20 end
```

It is after all only a demonstration program, nice and short, with just enough to show you whether it is indeed version #1 or not, but absolutely <u>any</u> file will do!

Now for the finesse. Put a formatted disk in drive #0 of unit #8 (or any drive of any unit, but these are the default numbers and keep the typing down), and type the following EXACTLY:

```
dsave"test  pgm"+chr$(160)+"ver  1"    (and  don't  forget
return)
```

Check your drive error led, and if clear, check the directory, which should show the following somewhere among the rest of the files:

```
1 "test pgm"ver 1      prg
```

You may now at your option verify the file, or dload it, demonstrating that the file responds correctly either to the 'short form' name of "test pgm", or it's true name, which is "test pgm"+chr$(160)+"ver 1"

Now for the fun part! Create a second program as below:

```
10 print"this is version number TWO!"
20 end
```

type the following EXACTLY:

```
dsave"test  pgm"+chr$(160)+"ver  2"    (and  don't  forget
return)
```

If you care, try the specific directory request which follows, which will list only the file(S!) specifically named "test pgm"

```
directory d0,"test pgm" (return)
```

Remember that this command excludes files named "test pgms" or "test pgm ver 1", and will ONLY list the file EXACTLY named "test pgm".

You may edit the program creating if you desire ver 3, or 4, or however many versions that it takes to get the dad-gummed program to work the way you want it to. Each except the first must be addressed in the long-form version to be exclusively selected for loading, but every version of your program is now preserved on disk, so when you brilliantly typed 'delete -60000' instead of 'delete 60000-', or replaced the old subroutine with a glitzy new one (that doesn't work), or revised and modified everything until you're hopelessly lost, there is an easy way to take a few steps back and find a version from before the brainstorm.

When a version is hopelessly defective, or demonstrates a unique grasp of the wrong approach, you can scratch it alone by using the appropriate 'long name' in a scratch command. i. e.:

```
scratchd0,"test pgm"+chr$(160)+"ver 17" (& return!)
```

Once development has been completed, and you are 100% certain that the previous 37 versions are no longer needed, you can perform all relevant housekeeping in two easy steps:

```
-renamed0,"test pgm"+chr$(160)+"ver  38" to "final  pgm" (&
return!)
-scratchd0,"test pgm" (& return!)
```

Gadzooks! 01,files scratched,37!!!

Frankly there are other ways to do this, particularly dsave"test pgm ver 1" being the least complicated, but how about files which are called by other files, either basic program chaining, or when bload'ing a machine language file for execution from basic, or just to keep your documentation straight when improving an already functional program. Before I discovered this technique, I either had to modify the calling program somewhere (remember don't fix what ain't broke!) to compensate for the 'newer and better' second program, or fall into the dsave"@..." trap, where a 99% functional version gets replaced by a 9% functional version @#$%²&&*!!!

Mr. Anthony J. Goceliak
32 Cottage Street
Jersey City N. J. 07306

MIXED CP/M-86 AND 'NOMAL' OPERATION ON A HARD DRIVE

By: Anthony Goceliak

For those of you who are doubly fortunate to own a 9090 drive and also run CP/M-86, the drive may be automatically partitioned by CP/M-86 so that (roughly) half the unit is available for CP/M and half for usual operation, with the two halves virtually invisible to one another. There is a rub, however. If you operate the drive in this manner, you can never 'collect' or validate the drive, since the cp/m blocks will all be freed and become exposed to overwriting on the very next 'save'.

I am a reasonably careful individual, but the prospect of having to sort out 3 and a half MEGAbytes of tangled files just because of one splat sent shudders through me. With a data space that large, you cannot take chances.

Consequently I wrote the following basic program, which, when properly run, puts an ampersand file on the hard disk which straightens things out after a 'collect', with no danger to CP/M-86.

The right way to do this is to first rescue from the hard drive any files of value, then enter CP/M-86 and ask it, through a program (command) such as 'dskmaint.cmd' to header the hard drive. When CP/M-86 is through, exit CP/M and dload my program "9090prot cpm.gen". Run it and answer the prompt to indicate what unit number the hard drive is. An ampersand file will be created on the hard disk which is vital.

Once this file has been placed on the hard drive, re-save all those other files you wiped out by the re-headering from CP/M-86.

You should NOT dsave the basic program to the hard drive for the following reason:

When you are 'in the bag', and are forced to collect the drive, you may NOT write to the hard drive without the possibility of exposing CP/M-86 or indeed one of your normal files to damage. Therefore, put the ampersand file on the hard disk now, before trouble strikes! Then, whenever you need to collect the drive, do so, but IMMEDIATELY follow the collect with the following:

```
open15,(unit number),15,"&prot*":close15 (and return)
```

The file takes only a few seconds to run, but it will prevent having to search literally for hours to see what files were and weren't damaged on your partitioned drive.

Mr. Anthony J. Goceliak
32 Cottage Street
Jersey City N. J. 07306

GUARANTEEING THAT YOUR BASIC PROGRAMS WON'T RUN UNLESS CORRECTLY LOADED

By: Anthony Goceliak

This article is written for several categories of CBUG'gers. First, for basic-language programmers who write programs and wish to 'idiot-proof' them; second, as a round-about way to detect whether someone is running Ms. Deal's Keytrix with which your own program is incompatible (round-about since it only senses the change in basic start address); or third, for someone who is running an important program and simply needs to know that nothing was mis-loaded. Be aware that on the b at least, some memory errors can yield an apparently correct dload with no abort, correct st and ds$, but a faulty copy of the program on memory. The bad copy is sometimes (NEVER say always) detectable via a verify"0[or 1]:program name",[unit number] -=- but exactly as I said in my article detailing how I reprogrammed my function keys to type this mess for me, unless you run that f-key magic, how many of you honestly verify each program you load?

Here is a simple addition to any basic program which will automatically verify the program you have just loaded into your b-128 low boy (hi-profiles who knows?), from whatever drive of whatever unit you have just dloaded (or loaded) it, AS SOON AS 'RUN' IS TYPED! (It also works if the program was 'shift/run'.) If a faulty copy has been detected, or if the original was written for the 'normal' b and you are running Keytrix, (or vice-versa) the program won't run. You get a simple (or fancy) message, and a chance to re-load before you have trashed your database due to line umpty-ump being wrong, or crashing your b because keytrix was overwritten by something.

The proceedure is simple. Add this line at or at least VERY near the beginning of your basic program, since it makes no sense to let a faulty copy execute bad instructions for only 1 minute instead of ten, does it?

```
1 bank15: qz=peek(544): qz$=right$(str$(peek(529)),1)+":the
name of your program" : verify(qz$),(qz)) : if st<>64 or ds
then print"Bad Load!":end
```

The characters 'the name of your program' must be replaced, and I won't insult your intelligence by telling you what they become, but please do not forget the colon or quote marks.

The fancy vs plain message can be done via the expedient of replacing the if clause with the following:

```
:if st=64 and ds=0 then goto "the line after the end of your
fancy message"
```

Once again, replace those nasty old characters including the 'double inverted commas' with an appropriate line number, but please don't get too fancy or just maybe the 'end' statement will be lost if the error was early in the program and your b will gallop off in pursuit of who knows what.

The program expects the values peeked to reflect the drive and unit the program were loaded from, so expect this to fail if you dload"program", and then scratch"old program",d1,u9 or header"new disk",d1,u14 before typing 'run'.

Last but not least, if you wish you may re-define variables qz and qz$ at will in line 2 or anywhere thereafter, or change them (zq and zq$ anyone?) if for instance your own program uses qz or qz$ and is used in a chain of basic programs.

Mr. Anthony J. Goceliak
32 Cottage Street
Jersey City N. J. 07306

QUICK TESTING COMMODORE 9090 AND 9060 HARD DRIVES

By: Anthony Goceliak

For those fortunate enough to have one of the CBM hard drives, data and program protection can be a non-trivial worry. The hard-drives are huge, 5 and 7 1/2 MEGAbytes respectively, with a directory that simply refuses to be full, leading rapidly to a drive just chock-full of all kinds of files, or at least so it seems. However, the more stuff that goes into the hard drive, the more that you worry about backups, or (shudder) loss of data.

This program will not help with data recovery, there are other programs and techniques for that, but it can alert you to cumulative deterioration in your drives performance, hopefully in time to do something about it.

The program is written in basic, with a section whose function is to create and activate an ampersand file tailored for your drives capacity (9060's are only 2/3 as big). This ampersand file is the real performance tester, the basic section in reality just creates the &file and reads it's results. What the &file does is to attempt a machine language read of each sector of each cylinder (track) on the entire drive, skipping nothing, whether allocated or not. The process takes roughly 5 1/2 minutes for the 9090 and roughly 3 1/2 minutes for the 9060. Only one attempt at a read is performed, and the results are tabulated and displayed by cylinder (track) so that you can determine whether the drive's performance is the same or (hopefully not), worse than the last time the test was run. If deterioration is noted, many times it will be evident through this test BEFORE a given sector has become unreadable with the normal number of re-tries on error, and therefore before the file has become trash.

Should this be the case, you will have an opportunity to temporarily store your critical files and re-header the hard disk to re-establish a benign repose for your information.

CBM also reccommends that this transfer and re-header be done if you physically move the hard drive (not from shelf to shelf, city to city). However, many times the proceedure is a big waste of time if the move was carried out carefully enough. Some hard drives need an hour and fourty minutes just to re-header, and this is only a minor fraction of the time some utilities spend in transferring your files back from floppies! Running this test after moving a hard drive can save a full day.

Using the program is simple enough, dload the 90xx tester and run. Answer the prompt to tell the program what unit number is the hard drive, and then let 'er rip. The results will be automatically tabulated and displayed in a form of how many single-try errors there were by cylinder (track), with non-zero results highlighted.

QUICK TESTING COMMODORE 80xx DRIVES VS. DISKS

The 80xx test pgm is a direct offshoot of the 90xx version, but since a floppy differs markedly from a hard drive since the disk is removable, the purposes of running a test like this are somewhat different. By using a 'benchmark' disk, you can rapidly determine if your drive has shifted alignment to a significant degree, or you can compare the alignment of your drives by running the test in one drive with a disk from another unit. If you are preparing to write to a disk which you didn't header, (a disk from a friend, or from CBUG) running this test first can indicate the relative likelihood of success and save you from a splat file and all the trouble which comes with it.

The program is run from your b. Dload the program, run it, and answer the prompts for unit address, drive number and number of passes. (Normally select one pass unless you wish to 'exercise the drive', a useful proceedure for uncovering heat-related troubles)

Upon completion of a test, the results are displayed.

Mr. Anthony J. Goceliak
32 Cottage Street
Jersey City N. J. 07306

A FASTER WAY TO BACKUP DISKS WITH AN 80XX SERIES DRIVE

By: Anthony Goceliak

The ampersand files described by this article will function correctly on any of the following drives:
8050 dos 2.5
8050 dos 2.7
8250 or 8250lp

Everyone knows that floppy disk backups are a wise precaution against all manner of ills, from accidental damage (coffee anyone?) or leaving a disk lie in the sun until thouroughly warped, to magnetic mayhem ranging from overwriting the wrong file or having the phone ring while sitting on top of your disk. The list is nearly endless, and I'm sure we agree in principle at least that backing up your disks is the smart way to go.

However, (there is always a 'however') backup is SLOW! My dos 2.7 8050 takes over SIX minutes and an 8250 requires over ELEVEN minutes to perform this routine task. Human nature being what it is, I never seemed to decide to backup until after disaster struck, when it was too late to do any good. After all, I had only added two or three letters or articles to a Superscript correspondence disk, or I only made a few changes to those Superbase files, or wrote one or two new programs in basic. Postponing a backup after one session isn't too bad, but the same logic prevails at each computing session, so while an active disk had(!) 630 blocks free, good 'ol backup seems to have 1800 blocks free, and when it is taken out to 'save the day', hmmm.....

ACTUAL TIMES FOR THE OPERATIONS

	8050 dos 2.7	8250lp	
backup d0 to d1	6:02.1	11:05	(normal dos command)
&reback 0to1+	1:35.7	3:12.1	(for minimal change)
&reback0 to 1	2:25.3	4:51.0	(unlimited change)
&certify bac+	1:35.3	3:11.2	(or less if the disks are NOT identical)

These times were recorded from the time that the drive began spinning to do it's assigned operation until the operation was completed successfully. If there are re-attempts made due to recoverable errors, the times will be extended by a modest amount. However, a properly aligned and functioning drive should yield results identical or quite close to these.

Consequently I wrote the four programs entitled "&reback.....". DOS's backup command assumes nothing, and so what it does is first grab the disk id#s from your source disk and then proceed to header the destination disk as though it were blank. This is necessary the first time that you use a disk, but assuming that the destination disk has not been subject to damage, is redundant ever after. My code eliminates this needless re-headering while testing that there is no damage to any of the 'invisible to the user' bytes which a full header creates but no ordinary disk operation ever changes.

Should there be any trouble detected, the drive will abort and show the error led red. DOS 2.7 drives, being endowed with a jump table, perform the proper set-up to send an error string message to the user if requested, but since there are different versions of DOS 2.5 around and no jump table to build a message, DOS 2.5 drives will simply refuse to do anything else for you until the disk is removed from the drive. Why? Because an error message will be built by DOS 2.5 when my code has been exited, and that message invariably says 00,ok,etc. (the last thing DOS 2.5 was told to do was load and execute my file, and it did that ok). Therefore, while a DOS 2.7 drive warns you of an error when used from within Superscript II or Superbase, DOS 2.5 would simply show 00,ok,... This method forces you to look at the drive if it is DOS 2.5 and become aware of a problem. Of course even with DOS 2.5 when there is no error, the code shuts down in an orderly manner and the drive responds completely normally.

DOS also copies a track more slowly than is possible, and my code implements a more efficient method of selecting which sector should go next. Note carefully however that the standard DOS proceedure of write and then verify is maintained, so if the code follows through to completion, you are assured that the backup was indeed valid and correct. There is no purpose to speeding up a backup by eliminating the verify, consider merely headering a disk without backing up, after all over 50% of the bits will be correct already!

WHY FOUR DIFFERENT FILES?

The first pair of files perform backups using drive 0 as source and drive 1 as destination, while the second pair reverse this, using drive 1 as source and drive 0 as destination. I firmly believe in labeling my disks with the drive and unit they were headered by and always writing to them using the same drive and unit. Having both pairs of &files enables you to do likewise.

The code within the two files of a pair is optimised for either minor change or a major rewrite. This allows you to select which file to use to most efficiently (read that as faster) backup a disk.

For minor changes, perhaps below 200 blocks of altered code, select the file named "&reback (s) to (d)+", where (s) is the source drive 0 or 1, and (d) is the destination 1 or 0. This file is optimised by directing it to first verify that the blocks are identical between source and destination. If they are, that sector is considered complete and no further action is needed. The program moves on to the next sector. If the verify fails, (a file was added, deleted or modified) then the source block is moved to the destination and a verify is once more performed before moving on. The code automatically accommodates itself to the size of your drive and 'knows' when to move to the next track and also when there is no next track (we're done!).

Disks which require more than a few hundred blocks of alteration will be more efficiently handled by the straight approach, which reads a sector from the source, writes it to the destination and then verifies the written sector.

The fifth file in this section is labeled &certify bac+. If you are ever in doubt as to whether two disks are exact backups or not, this file can help you. Run it, and it will report either 00,ok,etc... or show (with dos 2.7) the first sector which failed to compare between disks. Again DOS 2.5 will only turn the error led red.

USING THE FILES

Place my disk in drive #0 of the unit in which you wish to do the work. From Superscript II or Superbase go to the disk mode and type the name of the file you wish to use. For instance, if you wished to backup FROM drive #1 TO drive #0, and there have been only modest changes to the disk since you last backed up type:

&reback 1to0+ (and return)

From basic, assuming the same function was desired, type:

open15,(unit number),15,"&reback 1to0+" (and return)

Of course you must replace the (unit number) with the actual number, usually 8. (i.e. open15,8,15,"&reback 1to0+")

The files can be actuated from the machine language monitor as well if desired.

At this point, however you have gotten there, the action will be the same. The drive loads and begins to run my code. However don't panic. All it does so far is to blink the drive's led's at you to remind you to change disks. Insert the disk (in this case the SOURCE disk) in drive #1 and close the door. Remove my disk and insert the disk (in this case the DESTINATION disk) into drive #0 and close the door. In a few seconds the operation will proceed. The error led or ds$ message will inform you of the success or failure of the operation when it is through.

The &certify bac+ program is activated and run in the same manner as the rebacks, but since it never writes to a disk, the distinction between source and destination is academic, and only one version is needed.

When the operation is complete, SS II or SB will automatically show the ds$ message and close the command channel, but if you are using the files from basic, dont forget to type:

```
?ds$:close15   (and return)
```

If you have DOS 2.5, the message will be displayed when the operation was successful, but if it wasn't don't forget to pull the disks to restore control over your drive.

Common reasons for failure of the reback operation are:
1. use of an unformatted disk as destination. Solution: 'header' the disk using the same id# as the source.

2. destination disk has a different id# than the source disk. This is a protective feature, so that you cannot wipe out your superbase applications disk by overwriting it with your Superscript correspondance for instance. Solution: change to the correct disk and repeat the operation.

3. an unrecoverable disk error was encountered on either source or destination. These programs will re-try a given number of times in case of a bad read or write, but they are set to abort when the limit is exceeded. This indicates either an intentional or un-intentional real error on disk. The programs will NOT duplicate copy-protected disks.

If you generally do not pay attention to which drive a given disk was headered by, you may have minor alignment troubles, and the operation may proceed if you remove both disks, select the opposite direction (i.e. &reback 0to1+ instead of &reback 1to0+) and try again. Remember that the source disk must mow be in the other drive!

When you no longer have to wait quite so long for a backup, I hope that you will perform this vital function more often. Never forget that sooner or later you will need that backup disk, and the more current it is, the less work there will be for you!

Mr. Anthony J. Goceliak
32 Cottage Street
Jersey City N. J. 07306

DOCUMENTATION FOR THE B-SERIES DISK EDITOR "MR. ED"

By: Anthony Goceliak

This program requires a b-series computer (b-128 etc.) and one or more of any of the following drives in any combination:

2031
4040
8050 dos 2.5
8050 dos 2.7
sfd-1001
8250
8250lp
9060
9090

Several years ago, when I released the program which allowed a purchaser to install Ms. Deal's Pre-Superscript onto a dos write-defeated disk, some people began to call me a disk wizzard. Perhaps that was a bit premature, but with the benefit of this b-series specific disk editor, every one of you have the 'wand' to revive or trash any of your favorite disks.

The program was written by me and updated as necessary until there was literally no disk editing function which I ever (needed / craved / desired) which this program cannot do. Unlike any disk editor I have ever seen, this program does not care whether the internal disk id#'s match, whether you have swapped disks between the time you read a block and now wish to write it back, whether there is a valid directory or bam track, or whether you are reading from unit #9 and writing to a DIFFERENT drive type and unit, it will simply do what you tell it without interference from normal DOS error messages. You get the opportunity to see and potentially edit a block with a DOS error, even if the master dos version byte on the bam has been changed.

The program is extremely easy to use, since every single stroke option is displayed on screen at all times with only one exception. There will be more on that exception a bit later when I warm to the task of writing.

The program dloads in a normal, single proceedure, and when run it requests four pieces of information, namely the source Unit address, source drive Number, Destination Unit address, and destination drive Number. Source refers to the disk being read from, and destination usually refers to the disk which will be written to. Defaults are provided for Unit #8, drive #0 for both source and destination. However, you may type over these and change to any legal IEEE Unit address and either drive number. The Unit address does NOT have to be the same for source and destination, nor does the drive type.

Drives supported automatically by the program are the 2031, 4040, 8050 dos 2.5, 8050 dos 2.7, 8250, sfd-1001, 9060, and 9090. Maximum track, minimum track, and maximum sector allowed on either source or destination on a given track are automatically computed, and if the boundaries are exceeded, the program does not allow the drive to go off on a fruitless hunt for track #200 for instance. However, I never believe in artificial boundaries, and here is the 'unmentioned' option never displayed on screen. By typing the letter 'I' when selecting an option, all track and sector limitations are eliminated, and you may (profitlessly) send your drive off in search of track 255 sector 255. All the options were chosen mnemonically, the letter I stands for 'I know better' for those who must know. Seriously, the limits are removable in order to accommodate my 85 track mode for the 8050 drives, or a similar treatment allowing track 36 on the 2031 or 4040.

As soon as you have specified the source and destination information, the program begins by reading the first directory file block on the source drive, again automatically computed by drive type. The display is in a four row by 64 wide format, which allows rapid inspection of each filename when you are attemting to access and edit a given file. It is worth mentioning that the program will NEVER BUMP the drive head whether a block was read correctly or not, unless you specifically request it at the beginning of a run when the program cannot find the directory.

OPTIONS

Each option (with the exception of 'I know better') is always displayed on screen in a two column format, so the program is essentially self-documenting while in operation. Additional information displayed include the current source unit address and drive #, destination unit address and drive #, current cursor position displayed as a block cursor and in Both decimal and hexidecimal number systems as a position number, and the current value of the byte at the cursor position, once again in both decimal and hexidecimal numbers. Inputs may be given in either decimal or hexadecimal, and one of the options allows you the choice of your 'default' number entry system.

An <<abridged>> listing of the options available from within the program follow:

<<The full list is to be found to Mr. Goceliaks latest disk in this issue of the library>>

a = again. repeat the last command given.

b = begin. select a new block to inspect or edit by track and sector.

c = change. alter source or destination unit address or drive #.

d = dir. goto the first directory block on the source drive. (this is automatically compensated for regardless of drive type)

D = cp/m-86 dir (Capital 'd'). first cp/m-86 directory block, automatically selected too.

e = examine. repeatedly switch display between two blocks for visual comparison allowing you to 'blink compare' the blocks for differences.

f = fetch. fetch the block stashed by option 's' from the b-memory. (If the b is not powered down, a fetch may be made even though the program has been exited and even re-loaded.)

h = hex.	o = original.	v = view.	* =
j = jump.	q = quit.	(tab)	/ =
l = last.	r = rewrite.	(TAB)	+ =
m = move.	s = stash.	(ce)	- =
n = next.	t = text.	(CE)	

The four cursor keys and 'home' key each move the cursor as they are named.

FEATURES

If you have ever accidentally altered the 'dos version' byte on the bam, you have probably discovered that the disk has now become write-proof. Not with this editor, so any corrections needed can still be made, and the dos version byte can itself be restored, re-opening the disk to normal writing as well.

Bad sectors. This program allows YOU, not dos to judge when a sector has been hopelessly scrambled, and ALWAYS at least lets you see what the drive has been able to read. If the sector was incorrectly read, ten re-tries are automatically performed, but NO BUMP, and you will NEVER be precluded from seeing how good the read was. Many times a single byte may need correction, allowing a rewrite and complete recovery of a disk file! You may always, entirely at YOUR option, go again (another ten tries), accept a faulty read, or just forget the whole thing and try something else.

You may swap disks at any time the drives are not spinning! The disk id numbers are not an obstacle, nor is the drive's automatic close channel when a disk is pulled, whether you choose to change the source or destination disk, the program will perform exactly as intended. If you have one block which has been edited, which you wish to transfer to an entire series of disks, go right ahead, changing one after another, and selecting 'm'ove or 'r'ewrite as appropriate. If you have been adding this block to your 8050 drive unit #8 and would also like to add it to your 2031 (or 9090) drive unit #10, just select option 'c'hange units. After specifying the unit address and drive number of the new destination, continue rewriting or moving just as though nothing had happened.

Do you only own one drive? You can swap disks, reading from one and writing to another without regard for the disk id#, or you can even exchange two blocks on the same disk by using the 's'tash and 'f'etch options.

If you are unsure of the effects editing a single block will have, you may 's'tash the original block, edit it as desired, and exit the Mr. ed program!

As long as you do not disturb the b-series computer memory in bank 15 page six, (which is NOT used by basic, and only rarely by machine language) you can even try out the results of your editing without losing the original! Dload (if necessary) mr. ed once again and select options 'f'etch and 'm'ove to restore the block to it's original condition. If you power down the b-computer or crash it however, you will probably NOT recover your 's'tashed block.

If you are active in telecommunicating with your computer or in using cp/m-86 or ms-dos on the b-series with a co-processor board, you have undoubtedly learned that a conversion is required between CBM's version of ascii (cbmscii) and true ascii, used by most other computer makers. When editing, you may select the 'v'iew true ascii mode, which will allow you to read both capital and small letter ascii texts. This can be a great aid in finding the spot in the file you wish to edit! To prevent unwanted conversion of the actual disk file, a reverse video prompt warns you when you are operating in this mode. Disk writing is also inhibited to prevent the inevitable human error. However, if you know what you are doing, you may 's'tash a 'v'iewed (converted) ascii block, and then by selecting 'o'riginal cbmscii mode, 'f'etch the converted block, thereby allowing you to replace the ascii file with viewable cbmscii.

Ordinary editing of a true ascii file, as long as the 'v' option was not selected, is of course possible in the ordinary manner.

If you ever need to compare two blocks, the 'e'xamine option is just the ticket. When the option has been selected, you are requested to name a track/sector on the source drive, and a track/sector on the destination drive. Note that nothing is required to be the same. The program will show you first one and then the other block, "blinking" back and forth until you press any key to exit this option. It used to be a real pain to try to compare two blocks of machine language with each other, (m/l often doesn't provide for version numbers!), but I simply can't explain how vividly differing bytes stand out when using this f search.

In a similar vein, the option 'l'ast re-reads the last block specified. When you intend to compare a 's'tashed block with one on the drive, placing one finger on 'f'etch and another on 'l'ast gives you a semi-automatic version of the 'e'xamine option. This can be quite handy when the 's'tashed block either no longer exists on disk in the same form (having already been edited), or if you have swapped disks.

As I said in the beginning, this is the disk editor which I use, and it is capable of either repairing or completely trashing your disks. This is a POWERFUL tool, but exactly like a hammer it can either put things together or hopelessly smash them. The wisdom to use the tool correctly must be supplied by you.

Mr. Anthony J. Goceliak
32 Cottage Street
Jersey City N. J. 07306

CBUG West EASYWARE

EASYWARE = Easy To Use

Here we list newly diskovered (sorry) Easy-To-Use disks along with those we identified last time - for the sake of newcomers. Latest additions only are underlined.

Remember that the purpose of EASYWARE is to identify some of the CBUG library disks that are very easy to use. They aren't the only good disks, just the ones that we at CBUG West found easy to use right away. We just want to pass on some of our recommendations to you.

First come the disks that contain--Superscript files of information, articles or sermons. One of the things that is not always obvious when you obtain some of these disks is that you must load Superscript before these disks can be used. Not all of them contain annotated directories. They aren't always necessary on such disks, since the file names themselves usually give a good indication of the contents. Also, it is very easy to use Superscript to just browse through them and see what's there. Also, if you see a file named superspell.tm, ignore it - it just means that the disk provider used Superspell to check some file's word spellings.

8 Sermons #1
10 CBUG Print File
16 BASIC 4.0+ Tutorial Course - Instructions at the bottom of the Ready-to-print Version menu were a little confusing at first.
38 Summer 86 Part 1 Print Files
39 Summer 86 Part 2 Print Files - also given CBUG number 52
59 Winter/Spring 1987 Print Files
63 King James New Testament
64 Sermons 2 - Some files formatted for right margin 85
65 Sermons 3 - Ditto
76 The King James Bible (9 disks)

The following Superbase application disks were identified as EASYWARE. The only criticism we had for these disks also holds for most of the other Superbase application disks. Namely: the labels don't indicate that the user must first load Superbase before using these disks as the data disk.

49 Medical Finance #2 - Useful help files, but user has to directory the disk to find them
61 Super Teacher (Upgrade of #50) - Nothing told you how to get back to the Super Teacher Main Menu if any of the programs died/quit.

The following utility or application program disks are useful for the average user and have either annotated directories (a file on the disk identifying the purpose of all the files on the disk) or a SHIFT/RUN start up program.

3 Swan's Utility - Would be nice if the user could obtain a hard copy of the main start up menu.
28 Casey's Scrubber - Warning: Follow all instructions exactly!
36 London Sampler
37 SUPERPRINT - Good collection of programs
45 CBUG Utilities & Misc #3
47 dFile Database Program - Menu driven. Instruction files are in Superscript (with no indication).
58 Dittinger's Utilities - Needs more instructions
74 SuperOffice Scrubber - Warning: Follow all instructions exactly!

The following programs are for the more experienced user and/or programmer.

12 Scott's B-Mon - For the machine language programmer
27 Goceliak's Gold Mine - Lots of useful information for experts
51 JCL Workshop & Assembler - User must read all of the extensive documentation (true of any such large package).
75 Machine Language Index - This Superbase Application is for assembly language programmers. There is only one database on this disk, and the start up program (start.p) should choose it for you, instead of asking for the database name.
78 Gold Coast Instructional - Some programs do not return to the SHIFT/RUN menu (minor annoyance)

That's all for now. Each time we review more and more of the library disks available. So next time we'll have more EASYWARE disks to list.

CBUG WEST SCHEDULE & COMMENTS

CBUG WEST - OSMOSIS

Have you ever gotten stuck? "How do I do this in Superbase?" "My BASIC program is just too slow in this area. How can I speed it up?" "I want to make an intelligent data entry program for my Superbase application." "How do I get Superscript to do what I want it to?" Or one of many other famous last words.

We have some experienced programmers and engineers at CBUG West that are waiting to pass on their expertise to the novice and intermediate users. If you're any of these latter, or are an expert yourself, come to CBUG West for a no nonsense, low jargon, high tech informational exchange.

For the next few months we'll be examining ACTUAL SUCCESSFUL PROJECTS, done by experienced engineers and programmers to learn by their design decisions, whether toward success, or toward another "learning experience." We'll be using the B software that YOU use: Superscript, Accounting Software, Superbase, Calc Result, Word Result, JCL Workshop, and others.

THIS IS NOT A SUPERBRAIN FORUM with a mass of computerese and assembly language or machine language contortions. It is aimed where you are in your level of expertise.

YOU CAN GET CBUG LIBRARY DISKS FREE OR FOR $1 above cost!

Come join the growing group at CBUG West where we're deciding which CBUG library disks are EASYWARE. Show up and take home a disk for review and it's yours FREE. Also, anyone who attends any of the CBUG West meetings can obtain any of the disks reviewed that night for $1.00 (or $1 over the author's royalty for royalty disks).

Anyone can review a disk. Don't worry. The less you know - the better. We don't want experts to decide what disks are usable by us non-experts! Just grab a disk, follow the instructions on its label (if any), plug it in, and see if it makes sense to you. We'll help you, and we need your help.

CBUG West meets on the second Monday of each month at 7 PM at the First Congregational Church, 5th and Main Street, West Dundee, IL. Main street is State Route 72 (Higgins Road). 5th is about 6 blocks west of the Fox River and 2 blocks east of Illinois 31. Please disregard the Fox Valley Commodore Users Group meeting and go straight down to the basement where CBUG meets promptly at 7:00.

For information including meeting contents contact Warren Swan at (312) 665-1514 6 to 9 PM (please no later). For weather cancellation queries, contact either Herb Gross (312) 695-1316, or Warren Swan.

CBUG EAST MEETING SCHEDULE

October 23 - Hardware II - 8050 upgrade, SFD drive switches, etc.

Vern Kempfer will continue hardware demonstrations, showing how to replace the sometimes unreliable Zener reference diodes in the 8050 disk drive and how to install a switch on the SFD drive that will allow switching from 8050 to 8250 format. As time allows, other things Vern has done to his machines will be demonstrated. If you want to bring your drives and soldering irons (and a couple of diodes) to the meeting, Vern will help you do this relatively simple installation.

Nov. 27 - Library Demonstrations

By this time, a number of the programs in our library should have been reviewed. We will demonstrate some of the more interesting ones and/or some of those of most general use. Copies will be available for purchase.

December - No December meeting due to Christmas hollidays.

Jan. 22, 1989 - To be announced.

CBUG East meets at 2:00 pm, on the fourth Sunday of each month except December, although occasional rescheduling is necessary. In case of rescheduling, all on the local mailing list will be notified by one of these mailings. There are no dues for these meetings. No one even checks to see if you are a member of CBUG. There is a sign up sheet, to make sure that you will get these local mailings. We meet at Bethlehem Lutheran Church, Wesley Avenue and Greenwood Street, in Evanston. Greenwood Street is one block north of Dempster. Wesley Avenue is a block west of Asbury (a continuation of Chicago's Western Avenue). Enter on the north side of the building. Parking on the street and behind the church. If you need more details, call Marilyn Gardner at 866-9159.

A PLEA

Those of you who have been with CBUG for a while will recognize this request, and our gratitude to you for complying with it so far. We do have many newcomers since this was explained before. So here goes ...

When you write to anyone connected to CBUG and you expect an answer back, please include a Self Addressed Stamped Envelope with enough postage to get back to your country, state/province, burg/city, what-have-you, and door. <<see note 2 below.>> This may come as a surprise, but none of us connected with CBUG get paid even a penny for our efforts. In fact, this is a hobby that costs each of us anywhere from a little to a lot to provide what we do.

Furthermore, we're not asking or soliciting funds for services rendered. We'd just like to remind you to "Do unto others as you would have them do unto you." The courteous thing for the requester to do is to provide to the requestee, not gratuity, but just what is common sense. Namely, his or her expenses in the form of return postage and parcel.

I'm truly sorry, but I myself am sitting on some letters that were sent to me from remote parts of the globe, the answer to which would cost me a lot, only to say that I can't help the requesters (who didn't include a SASE). <<see note 1 below>>.

In advance I thank you all for your consideration. All but a very few of you have been faithful in your thoughtfulness. And a word to the wise is sufficient.

<<In lieu of SASE's and mailers, it may be more practical to just send sufficient postage stamps or dollar bills to cover the costs>>

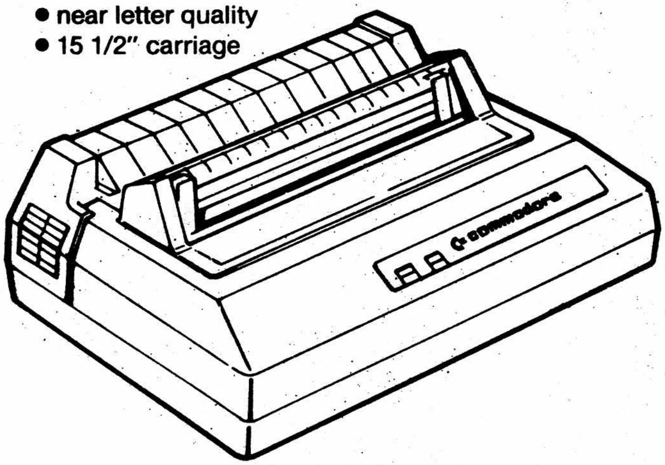
<<NOTE: CBUG maintains a modest postage budget for correspondence purposes. Considering the volume of mail we process, it is more convenient for us to NOT to handle SASE's and member mailers. Contributions to CBUG help make this service possible; those so inclined are requested to contribute at some convenient time (such as at renewal or with an order).>>

<<SPECIAL NOTE to non-U.S. members: There is a special insert in overseas copies of this issue explaining the new remittance requirements enforced by U.S. banks and also alternative methods of covering postage costs since foreign postage can not be used in the U.S.>>

Thank you,
Warren D. Swan

8023P 160 CPS Printer

- 100,000,000 characters
- near letter quality
- 15 1/2" carriage



PRICED AT $179 US
ADD $16.95 SHIPPING US

The 8023p supply is in an unpredictable position at the moment. Before we could place a large order with Commodore for these printers, they sold all 1,600 of them to a large import-export firm. At this time we have not been able to locate where they were shipped. The available supply of these may also be very limited.