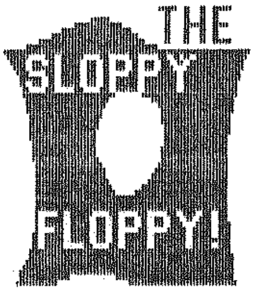
WEST PENN 99 ERS

VOLUME 1 NUMBER 3 FEBRUARY 1997



HOW TO FIX DISKS

By Nirej N. & Mike Ballman

This artical originaly appeared in the Spirit Of 99, the newsletter of the Central Ohio Ninty Niners. I am taking it from the Vast News.

Did you ever try to catalog a disk and find out the disk Contrroller thinks the disk is NOT initializd? But you know better! What do you usually do with a blown disk? Moost people delete the file giving them the problem. Usually that does correct the problem, but it also gets rid of that file forever. Thhe ultimate solution is to use DISK FIXER by Navarone Industries. (Editors Note: Another superb sector editor is the John Birdwells Disk Utilities.)

Here is the process to fix a blown disk

First acquire a DISK FIXER from a friend, or buy one, there worth it. Get a hard copy catalog of the blown disk, or even better, get a complete (old) catalog of what should be on the disk. If a complete catalog is not available try to remember what should be on the disk and write those names down on paper. Once you have a catalogof the disk, you are ready to start using the DISK FIXER.

Insert the DISK FIXER cartridge and select option 2 from the title screen. Upon doing so you should the DISK FIXER menu. Do the following if the most recent catalog of the bad disk tells you there are more sectors used/free than is logically possible >358 for single sided & 718 for double sided disks. For example, If the catalog list 500 sectors used/free on a single sided disk then do the following ELSE GOTO THE PARAGRAPH ON "sector one".

This part tells you how to fix up sector 8; which is the sector containing the information concerning the disk name and number of sectors used/free on the disk. If the disk catalog tells you the used/free sector information is in error then sector 8 needs to be fixed. The easiest way to do this is to copy a good sector 8 from another disk to the blown disk. Here is how to do that:

CONTINUED PAGE 4.

WEST PENN 99'ERS CLUB INFO

Next Meeting Date:	February 18, 1997
Meeting Location:	Penns Woods Civic Association Just off Route 30 N. Huntingdon, Pa
Time of Meeting:	7: P.M.

GENERAL ITINERARY OF OUR CLUB'S MEETING

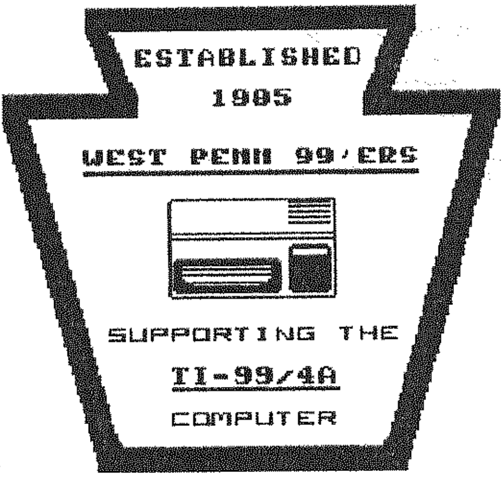
6:45 P.M.	Doors Open
7:00 P.M.	Genrral Meeting
7:45 P.M.	Demos and New Info
8:45 P.M.	Questions and Answers
9:30 P.M.	One on One Help
10:00 P.M.	Socializing
10:00 P.M.	Doors Close

MEETING HIGHLIGHTS FOR THIS MONTH

Submarine Commander..................	Demo by J. Wiegand
Computer War.........................	Demo by J. Wiegand
River Rescue.........................	Demo by J. Wiegand
Casino Solitaire.....................	Demo by Paul Brock
Open Intrest.........................	Demo by Anyone

LIST OF WEST PENN OFFICERS FOR 1997

President:	Paul Brock	412-478-2754
Vice-President:	Norm Rokke	614-264-6442
Treasurer:	Ed Mandich	412-824-5566
Recording Secretary:	Paul Brock	412-478-2754
Corresponding Secretary:	Paul Brock	412-478-2754
Librarian:	Mickey Cendroski	412-265-5201
Newsletter Editor:	Paul Brock	412-478-2754
Assistant Editor:	Paul Brock	412-478-2754



The West Penn 99'ers Users Group is a Non-Profit organization, dedicated to encouraging the continued use of the TI-99/4A home computer.

Our Membership Fee is:

* $15.00 per year for an INDIVIDUAL / FAMILY membership.
* $10.00 per year for a NEWSLETTER ONLY membership

Those having Full memberships are entitled to the many extra benefits our club has to offer.

Some of those benefits are:

* Getting to meet some of the nicest people.
* Demos of the latest TI-99/4A software.
* Free copying of our West Penn 99'ers Disk Library.
* Up date of T.I. news,Local,National,International.
* One on one help / Problem solving.
* Participation in our Module Lending Library.
* Participation in our Video Lending Library.
* Ribbon re-inking- for just $1.00 per ribbon.
* Various Computer supplies - at a substantial savings.
* Ability to trade or sell computer equiptment,or electronics.
* Help on getting equiptment fixed.

We meet the third Tuesday of each month at the PENNS WOODS CIVIC ASSOCIATION in North Huntingdon,PA. at 7:00 P.M.

If you can't make it to our meetings...at least become a Newsletter member - and enjoy our NEWSLETTER FORMAT- done entirely on a TI-99/4A computer.

SEE PAGE 10 FOR OUR WEST PENN MEMBURSHIP APPLICATION.

FOR THE RECORD BY PAUL BROCK

JANUARY MINUTES

I finally got the meeting started,it was around 7:20 when I got everything out of the box. The frist thing that I did was to welcome a New visitor that Jim Wiegand brought. She was his daughter. We had a nice turn out. (About 14), I don't know if I counted myself.

The corresponding report was not that great. not much to speek off. There wasn't any corrections to the minutes. There was a few misspelled words.

Now hold it right there,we did talk about the fact that B.C.99ER U. G. has sent their last newsletter. Ron Warfield has resigned as ED.. I for one have gotton a lot of info. from his N.L.. Thus in return I will correspond with with him and continue to send him a N.L. in return for their information sheet. I may need help in the future. What I understand the club is not going to fold. They are not putting a N.L. together. So we may still exchange ideas. All agreed.

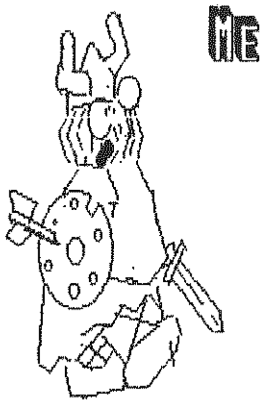
We still have a good report Treasurer. A job well done. To keep up with the work Art came with the refreshments and the re-inking machine. Norm got a spark going with the artical in Micropendum on C99. If there is enough intrest we will have a class on C99. So we may be having some classes again!

We officially have a new librarian. I got a volunteer, This was something that has been hard to do. Mickey has taken over as the new librarian.

Norm mentioned that he will be attending the FEST WEST. If anyone is in need of anything bring your ideas to the meeting.

I couldn't get the demo's to work the way I wanted them to work, so I asked the winner (Jim Wiegand) to DEMO them at the next meeting. See you on the 18th of February.

Untill then my QUILL has run out of ink!



MESSAGE FROM THE PRESIDENT & EDITOR

WoW!!! This is my third NEWSLETTER. I didn't think that I was going to get this far. My wife being sick most of the time, I really didn't think that I would be making any meetings let alone doing this. I hope I can continue. I can remember when I was in need of help. I still do!!

Speaking as president I wish to thank all those West Penn members that take home the TI equiptment and all the libraries this is a big help when so many have to travel so far. It is fun when working for the same idea. As always I am open for suggestions. Anyone can offer or suggest an idea that might help a TI'er will be taken with the most enthusiasm . If any member has a software that he likes and would like to demo it or have someone else demo just mention it at the meeting.

Norm Rokke mentioned to me that he would be traveling to San Jose CA., FEST WEST' 97 There will be FAFFLE TICKETS for a dollar each. Raffel prices include-

WHT SCSI Controller	WHT ATKeyboard	Super AMS
Horizon RAM Disk	TI CD-ROM	& MUCH MORE

Anyone wishing a ticket, bring to the next meeting a a mailing label to put on the stub! All the information is in the Nov. Dec. issue of Micropendum.

Normalcy is when you run out of money; insolvency is when you run out of excuses; bankruptcy is when you run out of town.

All guests are welcome,bring a friend or relative and show them what TI'ers do onthe third TUESDAY of every month.

HAPPY BIRTHDAY! TI'er where ever you are!

1> Insert a good disk in drive.
2> Read sector 8 of that disk:
R 8,1 [ENTER]
3> Put the blown disk in drive.
4> Write good sector 8 to disk:
W 8,1 [ENTER]

If you catalog the bad disk, you will see that the disk name and the used/free information is the same as the good disk but do not let that alarm you. The bad disk is atleast partially restored to normalcy. Now we need to fix up the blown disk as much as we can. This is done by changing sector 1. Here is how to fix sector 1. First, get the most complete catalog and the most recent catalog of the bad disk in front of you. Thhen compair the two catalogs to see which filenames are missing. Next, compile an alphabetical list of all the filenames which are and should be in the catalog. Thhen you need to find the corresponding sector for each filename. This is done by using the FIND STRING function of t6he DISK FIXER:

1> Put the bad disk in drive.
2> Find a filename by:
F 8,208,1 [ENTER]
Type in the filename [ENTER]
3> Ignor the **ERROR IN SECTOR** message
4> Write down the sector number for that filename.
5> If that filename could not be found
make sure you type it in correctly
and try again; otherwise that file
does not exist on that disk.
6> Repeat the process from step 2 for
all of the filenames.

You should have an alphabetical list consisting of two columns: Filenames and sectors. With that information in hand you are ready to begin fixing the bad disk. this is done by modifying Sector 1 of the blown disk. First you have to read Sector 1 from the bad disk by doing this:

1> Put the bad disk in drive.
2> Read Sector 1 of the disk bt:
R 1,1 [ENTER]

Then you want to alter the contents of Sector 1. This is done by using the ALTER function of the DISK FIXER. This process is best learned by observing a concrete example.

Lets say the blown disk has 14 files (**filenames**)on it. Thus there should be 14 entries on sector 1; one entry for each file. The rest of the sector should be all zeros.

Lets alter sector 1:

1> keep the bad disk in drive.
2> Enter the ALTER function:
A 0 [ENTER]

CONTINUED PAGE 5

CONTINUE from Page 4

3> Type in the following just as shown,
including the spaces:
1 2 3 4 5 6 7 8 9 A B C U E
4> Do not press [ENTER] yet!
5> If you saw a non-entry after the
E entryin the first column then type
in [0] and a [SPACE] and repete untill
the first column shows a zero.
6> Press [ENTER]
7> Write the revised Sector 1 to the bad disk:
w 1,1 [ENTER]

You have just entered a table of pointers to the files on the disk. The table points to the corresponding sector for each file name. This is the table that is updated aand sorted if you add/delete files to the disk.

Leave the DISK FIXER by typing [Q] for QUIT and press [ENTER].

Then catalog the disk. Lets call this new catalog the mixed catalog. Notice how the catalog is NOT in alphabetical order. It does however contain all of the filenames that you hoped and prayed would be on the disk! The next step is to alphabetize the catalog on paper and carry along the approiate sectoe number oof each filename. Here is an example of a mixed catalog.

MIXED CATALOG		SORTED CATALOG	
FILENAME	SECTOR	FILENAME	SECTOR
CAT	1	APPLE	E
SCREEN	5	CAT	1
VOTE	2	DEMO	7
FIRE	6	FIRE	6
APPLE	E	HELLO	9
HELLO	9	JUSTIFY	D
SCROLL	C	LOAD	3
LOAD	3	LOGO	A
TIME	8	PLOT	B
DEMO	7	QUICK	4
QUICK	4	SCREEN	5
JUSTIFY	D	SCROLL	C
PLOT	B	TIME	8
LOGO	A	VOTE	2

(ED NOTE) PHOTO COPY MAKES GOOD REFERENCE

CONTINUED PAGE 6

CONTINUE from Page 5

The example from page 5 shows how you should alphabetize the filenames and the corresponding sector numbers on paper. If you are unsure when dealing with funny characters, the system alphabetizs by lower to higher ASCII values. These values can be found on your TI Basic reference card. Once you have done this you are ready to enter this information into Sector 1. You donot have to enter the filenames, just the sector numbers.

Here is how to do that:

1> Put the blown disk in drive.
2> Read Sector 1 by entering;
R 1,1 [ENTER]
3> Enter the alter function;
A 0 [ENTER]
4> Type the sector numbers in the order
as shown for the above sorted example
catalog. Separate each number bby a space:
E 1 7 6 9 D 3 A B 4 5 C 8 2
5> Then press [ENTER]
6> Write revised sector to disk:
W 1,1 [ENTER]
7> Put a Weite-Protect tab on the disk:

You have now fixed up the disk. For verification quit the DISK FIXER program and catalog the disk. You should have no problems during the cataloging process. But you are not completely done yet! DO NOT add/delete any files or programs to this disk!

Get a fresh disk and initialize it to the same configuration as the blown disk. Then back up the blown disk to the fresh disk. Then catalog the fresh disk and you will see that used/free sector information is not correct. Thus, the fresh disk is now your working disk and the blown disk is now a disk for your archives.

Keep the blown disk in a safe place just in case you remember a file that was not previously recovered from the blown disk, Go through the above procedures to recover that new-but-old file.

HAPPY FIXING!

SOME FUN> Mechanic to customer:"According to our diagnostic computer, your car wants to have $800 spent on it."

C for yourself

Norman Rokke

When C99 first became available at the user group I got a copy with the intention of learning about C. However, when I couldn't get a running program from the sample programs, I put it aside and concentrated on other programming interests. Some years later I took a course in C but I never returned to C99. Recently I had an idea for a program I wanted to write and began writing it in Extended BASIC. I got a portion of the program written but it became obvious that Extended BASIC was not the most appropriate language.

One of the problems was speed. Because of this and other deficiencies of Extended BASIC which I won't go into now, I decided to take another look at C99. Again, when I tried to get running programs from some of the sample programs, I experienced difficulties. This time however I finally managed to figure out what I was doing wrong and was able to get the sample programs to run. I have been able to code a considerable portion of my program in C99 and am very pleased with the language.

In this article I want to explain how C99 can be used to create a simple program, just to show the overall process from start to finish. This process consists of four steps.

1. Using an editor to create the C99 program.

2. Using the C99 compiler to compile the C99 program and produce assembly language source code.

3. Using the assembler to produce executable code (EA option 3).

4. Creating EA option 5 executable code.

Let's look at a simple C99 program that prints **Hello world** on the screen. I've read that one of the developers of C wrote a program to do this as the first program ever written in the language.

As I describe how to carry out each step I'll assume you have the set of C99 disks offered by Vern Jensen in Micropendium. I'll also assume that you have a two disk drive system and that the Funnelweb disk is in drive 1 and that you will save your C99 programs and any other necessary files on drive 2.

To create a C99 program you use an editor. From the menu choose 1 PROGRAM ED. This editor is essentially the TI-Writer or Funnelweb editor modified so that it does not cause problems with the C99 compiler. I assume that you are familiar with the basic commands of the editor. Use the editor to type in the following program.

```
extern grf1(),printf();
main()
   {
   grf1();
   printf("Hello world");
   }
```

After you have typed the program in, save it with the file name DSK2.HELLO;C using SF. The ;C ending of the filename is a convention to make it easy to recognize this file as a file which contains C99 program code. Then leave the editor and go to the menu screen.

Before describing how the program is compiled, I'd like to briefly explain the above program. The first line

```
extern grf1(),printf();
```

provides the compiler with information it needs to properly handle the program you are writing. Much of C99 is contained in functions which are stored in library files. grf1() and printf() are two such functions. The code for providing the capabilities of these two functions is in separate files which are part of the C99 package. The grf1() function puts the computer in graphics mode (24 rows, 32 columns) rather than the default text mode (24 rows, 40 columns). The printf function works much like the PRINT statement of Extended BASIC. In order to use these functions in our program, we must tell the compiler that we are going to be using these functions which are outside of or *external* to our program. This statement also requires that at the appropriate time we must provide the necessary library files so that everything will work properly. We'll see later when and how that is done.

Every C99 program consists of one or more functions. These functions are like subprograms in Extended BASIC. The second line

main()

identifies the one function which **must** be present in every C99 program, a function whose name is main. The parentheses after the name identify it as a function.

The other functions in a program are either library functions or functions created by the programmer. This program does not use any of the latter. The most important thing to know about the function main is that the program begins by executing the statements in this function.

The { on the next line marks the beginning of a block of code for the function main.

The next line

grf1();

sets the computer in graphics mode. Note that statements in C99 end with a semicolon. This applies to the **extern** declaration statement above also.

The next line

printf("Hello world.");

uses the printf function to print something on the screen.

The } on the last line marks the end of the block of code for the function main.

Now let's compile the program we've written. Choose 4 C-COMPILER from the menu. Press ENTER to accept the default values for each of the three questions. Next press ENTER to accept HELLO;C as the input file. Change the name of the output file to DSK2.HELLO;S and press ENTER. The ;S ending of the file name is a convention to help you recognize this file as an assembly language source file. Now press FCTN-6 to start the compilation process. Type N in response to the question of whether you want to compile again.

If the compilation completes and the message indicates zero errors, you are ready to go on to the next step. However, if there are errors, you must go back to step one and edit the C99 program to correct the errors. If errors are reported use PROGRAM ED and make sure your program is identical to the one above. Then save the file and try the compilation again. If your file is exactly like the program above it will compile successfully.

The third step involves assembling the code produced by the compiler. Choose 2 ASSEMBLER from the main menu. Press ENTER twice to accept DSK2.HELLO;S as the source file and DSK2.HELLO;O as the object file to be created. The ;O ending is a convention to help you recognize the file as an Editor Assembler option 3 object code file. You can press ENTER twice more to accept the default values for the last two prompts also. Press FCTN-6 to begin assembly. The assembly should go without a hitch and you should get a report of no errors. Press ENTER to go back to the main menu.

Finally we have a program that we can run. To be more precise, it will run if we load the additional files which are necessary. Every C99 program needs the file CSUP in order to run. This file contains code which sets up the default text mode before the program runs. It also contains code to cleanly end the program when the program is finished. Copy the file CSUP from the C99-LIBS disk onto the disk containing the program files we have been working with.

Since we used library functions in our program, we also need to supply the files that contain those library functions. the function grf1() in is the library file GRF1 and printf() is in the file PRINTF. Both of these files are on C99-LIBS. Copy both of them to the disk with the program files and CSUP.

Now we are ready to run the program. Choose 3 LOADERS from the menu. Then choose 4 LOAD/RUN (E/A). Enter DSK2.HELLO;O as the filename and press ENTER. When the cursor returns press FCTN-3 and enter the filename DSK2.CSUP and press ENTER. When the cursor returns press FCTN-3 and enter the filename DSK2.GRF1 and press ENTER. When the cursor returns press FCTN-3 and enter the filename DSK2.PRINTF and press ENTER. When the cursor returns press FCTN-3 and press ENTER.

You will see a screen filled with DEF Table entries. Use FCTN-D to place the cursor on the **S** of **START**. Then press FCTN-6 and the program will run. Although the program doesn't do anything impressive it should be satisfying to see that what you have written actually produces a result. As far as I know there is no way back to the main menu from here so press FCTN-=.

As satisfying as it may be to get the program running, I'm sure you will agree that there has to be an easier way to run the program than what we just went through. There is.! What we must do is create an E/A option 5 program. In order to do this we need three more files on our program disk. Copy the files C99PFI, C99PFF, and SAVE from the C99-LIBS disk to your program disk.

C99PFI contains the DEFs SLOAD and SFIRST which are needed by the E/A SAVE utility. C99PFF contains the SLAST DEF also needed by SAVE. SAVE is the E/A utility program used for creating program image (E/A option 5) files.

Choose 3 LOADERS from the main menu and then choose 4 LOAD/RUN (E/A). The order in which the files are entered is critical so enter them just as listed below.

Enter filename DSK2.C99PFI and press ENTER. When the cursor returns press FCTN-3 and enter filename DSK2.HELLO;O and press ENTER. When the cursor returns press FCTN-3 and enter filename DSK2.CSUP and press ENTER. When the cursor returns press FCTN-3 and enter filename DSK2.GRF1 and press ENTER. When the cursor returns press FCTN-3 and enter filename DSK2.PRINTF and press ENTER. When the cursor returns press FCTN-3 and enter filename DSK2.PFF and press ENTER. When the cursor returns press FCTN-3 and enter filename DSK2.SAVE and press ENTER. When the cursor returns press FCTN-3 and press ENTER.

You will now see a screen filled with DEF Table entries. The entry we want is SAVE and it is not on this screen. Press ENTER to get another screen of DEF Table entries. SAVE isn't on this screen either. Press ENTER to get to the third screen of DEF Table entries. SAVE is the last entry. Use FCTN-D to position the cursor over the S of SAVE. Then press FCTN-6.

At the prompt for the filename enter DSK2.HELLO. Then press ENTER and let the SAVE utility do its work.

When the program finishes press FCTN-9 to go back to the LOADERS menu. Select 3 PROGRAM (E/A). Enter DSK2.HELLO and press ENTER. You now have your simple C99 program in a much more easily runnable form.

I use the E/A option 3 program to test for errors I don't make a program image file until the program is rather well debugged.

I hope that you find this introduction to C99 useful. That's all for now. C you next time.

User groups may reproduce this article provided they include the name of the author and acknowledge the West Penn 99er's Newsletter as the original source.

WEST PENN 99'ERS
C/O Paul A. Brock
General Delivery
North Apollo PA 15673-9999

Newsletter Editors

Please note new address
and up date your mailing list

FIRST CLASS MAIL

MICKEY CENDROWSKI
100 PINE STREET
RUSSELLTON, PA 15076
USA

FIRST CLASS MAIL

This newsletter was composed in it's intrety
using a TEXAS INSTRUMENTS TI-99/4A computer

NEXT
MEETING FEBRUARY 18TH. 7:00 PM

WP MEMBERSHIP APPLICATION

Name____

Address______________________________________

City______________State__________Zip__________

Please check one

☐ Newsletter membership
No library privilages
$10.00 Per year

☐ Full membership
Library privilages
$15.00 Per year

Complete and return to

Ed Mandich, 304 Chicora St. East McKeesport,PA.15053