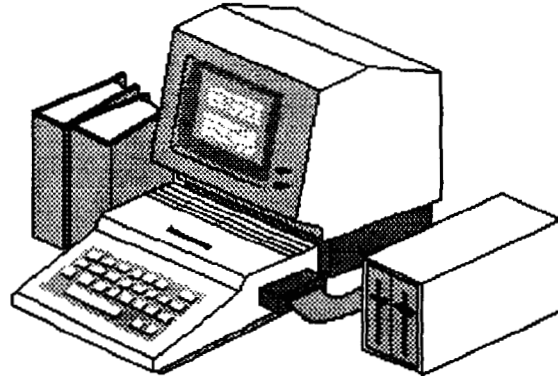# CoCo 123

March, 1995
Volume XV, Number 3

ThePresident's /term

by Eddie Kùns

Some weeks later, Eddie was sitting at his terminal at home (after logging in at work) when he suddenly realized -- it's March and I haven't written an article in two months! OK, I hear the whips behind me and feel the fire under my ... chair. Yes, it's time to talk to the Glenside community again.

As I reported last time, the catch phrase is delegate, delegate, delegate. With many, many thanks to the members of Glenside, business has usual has been going on. The Board of Directors has finally approved a new set of bylaws to go with the new constitution as Glenside takes steps to become a formal organization. No, don't worry, 'fests aren't about to become evening dress / white-tie-and-tuxedo events, and you don't have to pull your smoking jacket out of the closet. (Just what is the point of a smoking jacket, anyway? Ah, never mind.) However, we've been working for two years to formally set down in the bylaws the way the club operates. We have finally done this. The bylaws are far too long to include in the newsletter; if you would like a copy of the bylaws and/or constitution, watch this space for information on how to get a copy. Next issue I will tell you where to send your SASE.

With the new bylaws, the three Glenside vice presidents now have specific titles and have been elected to these titles within the board of directors. I would like to offer congratulations to Rob Gibons, Senior Vice President (he will help me coordinate the day-to-day operations of Glenside); Tony Podraza, Vice President of Special Events (he is in charge of the 'fest and is doing fantastic work keeping everything moving); Tom Schaefges, Vice President of Demos and Reviews (he will chair the second half of each Glenside meeting, introducing the demo, and will help gently encourage demo-ers to write a short writeup for the newsletter).

Another bit of success: I thought it would be fun to take pictures of the club leadership to publish in the newsletter. My favorite Christmas present was a new 35mm SLR camera, so this was an excuse to play with my new toy. Well, I took and developed the pictures and then passed them to club member and former vice president Gene Brooks. His sons scanned the pictures into computer format, and you will hopefully see our pictures in the newsletter in the near future.

I'll add a little bit of information about the upcoming annual last CoCofest: remember that when you reserve hotel rooms, mention "Glenside" or "CoCoFest." The number of paid venders is already growing rapidly, so this looks to be a good show -- I hope you all can make it. I will have more information about the CoCoFest for the next newsletter.

Here is Tony Podraza, Vice President of Special Events, as photographed by our Fearless Leader. Well, not exactly "as photographed," don't they teach Composition in Physics class, Eddie? But a little post-production work, a new background, and voila! I'm ready for my new career at the National Enquirer! Now, just a little more work and I'll give him more hair....

CoCo 123 Information

The Glenside Color Computer Club of Illinois is a not-for-profit organization whose members share an interest in the Tandy Color Computer ®. The CoCo 123 is the official newsletter of the Glenside Color Computer Club. The Glenside Color Computer Club of Illinois has no affiliation with Radio Shack ® and/or the Tandy Corporation ®. The opinions expressed by the authors of the articles contained within this newsletter do not necessarily reflect the opinions of the Editor, the Executive Officers, or the actual club membership.

We are committed to publishing a minimum of four issues and a maximum of twelve issues per calendar year. For a fee of $15.00, for January through December, you can become a GCCC member with full membership privileges. Send your dues to:

GEORGE SCHNEEWEISS
RR #2 Box 67
Forrest, IL 61741-9629

Here is a list of the 1994 Executive Officers and how to contact them. The club has four strong SIGs, Tandy Color Computer 1, 2, 3, and OS-9. If you have questions concerning these computers or OS-9 call one of these officers.

POSITION	NAME	PHONE	PRIMARY FUNCTION
President	Eddie Kuns	708-820-3943	The buck stops here
Vice-President	Rob Gibons	708-916-1118	Meeting Planning, etc.
Vice-President	Tony Podraza	708-428-3576	" "
Vice-President	Tom Schaefges	815-337-0117	" "
Secretary	Sheryl Edwards	708-344-5632	Records and Reporting
Treasurer	George Schneeweiss	815-832-5571	Dues and Purchasing
Editor	Mike Warns	708-627-9561	Newsletter coordinator
Telecom	David Barnes	708-587-9820	Club BBS SysOp
Printer	Dennis Devitt	708-629-2016	Newsletter Exchange
Advertising	Bob Swoger	708-576-8068	Newsletter Ads

CoCo 123 Contributions

If you would like to contribute an article, upload a file to the Newsletter Submissions section of the Glenside's Cup of CoCo BBS. Ditto for ads and news about your group. Please include a name and an address or phone number. If you would rather submit your article on diskette, they can be accepted in ASCII form and unformatted in the following formats:

RS-DOS 5-1/4" SSDD, 35 tracks
OS-9 5-1/4" SSDD, 35 tracks
MS-DOS Any standard format

Mail your articles to:

Mike Warns
31 South Edgewood Avenue
Lombard, IL 60148

Submission deadline for the CoCo 123 is the 15th of each month. Articles submitted after the deadline will appear in the next issue.

CoCo 123 Newsletter Exchange

The Glenside Color Computer Club of Illinois is pleased to exchange newsletters with other Color Computer or OS-9 user groups at no charge. Send your newsletter to:

Dennis Devitt
21 W 144 Canary Road
Lombard, IL 60148

Reprint Policy

If you desire to reprint any articles that appear here, please provide credit to the author and this newsletter.

We encourage your user group to copy this newsletter and distribute it at your regular meetings to your members free of charge as we believe that this will encourage better meeting attendance. If you are a user group that feels as we do, please let us know and send us your newsletter. We will then distribute it to our members and keep our attendance up.

Glenside Club Meetings

The Glenside Color Computer Club meets on the second Thursday of each month from 7:30 to 9:30 PM at the Glendale Heights Public Library. A social get-together always occurs afterward at the Springdale Restaurant.

Future meeting dates:
April 13, 1995
May 11, 1995

These are your BBSs. Please support them.

Glenside CoCoRama BBS
708-587-9837
9600 Baud

SandV BBS
708-352-0948
9600 Baud

Glenside's Cup of CoCo
708-428-0436
2400 Baud

Chi-CoCo BBS
312-735-3355
14,400 Baud

Glenside East
708-632-5558
9600 Baud

January Meeting

by Sheryl Edwards

The January 12th meeting was called to order by Brian Schubring at 7:50. The introductions started and we found we had two visitors, Richard Bair and Mike Carey. Mike Carey was from out of town--way out of town! How about from Prospect, Tennessee? He caught a glimpse of us at the Atlanta Fest. Being in the neighborhood tonight, he stopped by. While the rest of the intros were going on Bob (Gator) Swoger was working on the extra newsletters in the back of the room, with helpers sneaking back there now and then.

Brian's last iten of business Brian was transferring the gavel to 1995 president, Eddie Kuns. The official time of transfer was 8:05. Eddie started as new president with old business, just like a pro.

Dues need to be paid to George Schneeweiss as soon as possible. If "95" appears next to your name on the mailing label you are paid through 1995. If a "94" appears next to your name, you need to send in your dues to George (his address appears on the back of the front page.

There were questions brought up about PCM magazine. Does anybody know if it is still in business?

Mike Knudsen had some info on a 6809 processor and some extras. See Mike if you are interested in learning more about that.

Rob Gibons spoke about the CoCo 2 emulator for IBM compatibles. It can be downloaded from the SandV BBS (support your club BBSs!) or Compuserve. See if you can upload it to other boards.

Gator brought up that we should all call the club BBSs more often. This will help us to maintain communication among the members between meetings. He ahs made his board easier to use.

Bob brought up another good point: we need more articles for the newsletter. If you see or hear something interesting send it into the newsletter so we can all see it. Also, let's start running want ads to start trading or selling between us, in and out of state.

The next official order of business was new business. Eddie decided to take pictures of all of the new officers plus a group photo. I hope that these will end up in nice places, like the newsletter and the BBSs. *(Editor's note: Can't publish what isn't submitted!)* There are a few officers...Okay, just one...who was not prepared. Bad hair day. No makeup. Wrong clothes. Well, you know how that goes! Be nice to my picture!

Then we decided to take a break after the pictures, but, as we were breaking up, Paul tried to sneak in, but some of us caught him. (grin)

After the break came the demo. Mike Carey demoed CoCo MIDI Pro, a program with MIDI in-out capabilities that runs under RS-DOS on a 512k CoCo 3. He prefers it to Ultimuse. *(Another interjection by the editor: It was interesting to see Mike voice his objections directly to Mike Knudsen, who wrote Ultimuse. This kind of give and take is something that I find exciting about belonging to Glenside!)*

What's News?

by: Brian Schubring

'As the grand sounding music faded and the lights can up, the audience gasped at the player of such music as they had just heard. There on the stage, atop an ordinary table, beside a rack of musical instruments, sat a quaint little machine called the CoCo. Everyone stood up in amazement at the sight of this... this marvel that could do more then was understood, or even believed!'

This was an excerpt from a newsperson reporting on the amazing concert at the Chocolate Auditorium in New York.

'The power of the performance was more then was expected. I couldn't believe my ears after my eyes saw where music had come from.' Exclaimed a befuddled Bill Gates, the recipient of complimentary tickets.

The previous text was of course a fictitious byline from this authors mind . . . But who knows, maybe its possible. But the music part with the CoCo is not a dream. It of course is here and alive with the aid of such programs as Lyra, and UltiMuse3 from Lester Hands and Mike Knudsen respectively. But there is one other item that can be added to make the sweet sounds of CoCo even better. The MIDI Pack. Why would it be better? Let me count the ways.

1) Better thru-put. Better for busy music. (Lots of notes)

2) Both MIDI IN, OUT, AND THRU! (A plus for big synth systems)

3) RS-232 port (WHAT???)

Yes, I said an RS-232 port. But there isn't one on the midi pack, you say. That is so. But here is the breakthrough. Cecil Houk from Rulaford Research, has released the rights to build his version, (plus some added mods) to the GlenSide Color Computer Club. With his nod to proceed, I have taken the liberty to make some mods.

Alas, I couldn't do any of this without the help of Mike Carey from Prospect, Tennessee. He is what I would call a Midi power user, using the CoCo. I had hooked up with Mike at the last Atlanta CoCoFest. He has been a fast friend since then and co-conspirer of this venture. This venture is still in its infancy but with luck should be done by the end of this year or earlier.

The cost? Well, the club has decided to foot the bill to make a preliminary number and sell them for cost + some(?) and postage. The objective for this project is to make available to all with CoCo's and want music can have the best and the most for their buck and/or needs/wants. Look for future reports here. So-Long, FareWell.

Fest Information

by Tony Podraza

DATELINE: March 18, 1995

This is the third year that GLENSIDE Color Computer Club is UNDERtaking the sponsorship, planning, and execution of the Chicago-area held CoCoFEST! Last year brought approximately 24 exhibitors to the show.

This year, GLENSIDE is officially announcing that all tickets for the FEST! will be $5.00. For students between the ages of 6 to 16, there will be a 50% discount, when accompanied by an adult, and that children 5 years old and under will be admitted free of charge.

As of this writing, there are eighteen official exhibitors. They are, in no particular order:

1) Fat Cat Publications, Alan Sheltra

2) JoTA Productions, Timothy Johns

3) DigiGrade Productions, Dave Pellerito

4) DALTRUG, represented by Lee Veal

5) Farna Systems, Frank Swygert

6) Al Dages/R.C. Smith

7) DISTO, Tony Distefano

8) Budgetware, Brian Kitt

9) Adventure Survivors, L. E. & Nan Padgett

10) The Chicago OS9 User's Group

11) StrongWare, John Strong, et al

12) SBUG, Andre Levalle

13) Glenside Color Computer Club of Illinois

In addition, there have been verbal intentions of attendance by CoNect, Sub-Etha, Northern Xposure, The OS9 User's Group, ColorSystems, JWT Enterprises, Wittman Computer Products, and one or two others whose names escape me at this time.
{* As of press time, these have been confirmed.*}
Tony

Seminars are being planned for, with speakers including Allen Huffman; Curtis Boyle on NitrOS9; John Strong; Ed Kuns, Glenside President on State of the Community; and Brother Jeremy. However, exact times are not yet cast in concrete, except for the first meeting on Sunday Morning, which we are continuing as a tradition. That meeting will be the Meditation, Praise, and Worship Service presided over by Brother Jeremy, a CoCo Community member for as long as I can remember, having seen him at the first Rainbowfest I attended, which I believe was in 1986 in Schaumburg, Illinois.

The site of the FEST! is fairly easy to get to, being alongside of I90 at IL RTE 31, with a couple of turns onto West River Road. Air travelers can reach the site from Chicago's Midway Airport by taking I55 west to I355 north to I90 west to Elgin; O'Hare arrivals would take I190 out of O'Hare and follow the signs to I90 west to Rockford, but be sure to get off at IL RTE 31 in Elgin; Arrivals at Mitchell Field in Milwaukee will want to head south on I94 to I294 south to I90 west (same as before); and finally, should you fly into Rockford, head north to IL RTE 20 east to I90 east until you approach IL RTE 31, and follow the earlier directions to the Holiday Inn.

All in all, we at GLENSIDE expect to have a fun-filled weekend filled with friends, food, excitement, and Q-ZAR Laser Tag....oh, did I forget to mention Q-ZAR? Well, you'll have to come to the FEST! on Saturday @ 2:00 PM to find out about that. And rest assured, there will be a space for your name on the Badges that we forgot about, last year. BELIEVE ME!!!! After all the requests for a space for names on the BADGES! We won't forget them a second time! Come and join us in the revelry.

Now, the surprise.....some of you received RED newsletters in January. IF YOURS WAS RED, BRING IT TO THE FEST! You can only claim your prize there. It must have your mailing label on it. Come Saturday; come Sunday. Doesn't matter which day, but you must redeem your randomly generated selection as a prize winner at the FEST! (It was really random..I gave the mailing labels and the newsletters to 5 different people with both colors of newsletters and didn't tell anybody what was the reason for the red ones.) You MUST bring the Newsletter!!!!!

One-Liners

by Len Zielinski

ASCII a stupid question - get a stupid ANSI!

I parked my harddisk - and got a ticket!

Do not attempt to traverse a chasm in two leaps!

Okay, it's a long story, and I don't understand it,
but it might help YOU!

Arrgh!!!!! My disk crashed, now what do I do?

by Dave Gantz

Fixing crashed disks or at least recovering data from crashed disks can be a time consuming ordeal which is why the best recommendation is to ALWAYS keep current backups or DON'T use the disk at all if it is your only copy. However, crashed disks do not have to mean lost data if you know how to go about recovering the data.

That's what this article is all about and comes as a result of the three crashed disks I've had this past week as well as Chris Spry's current predicament. I have co-written one other article on this topic, but that pertained more specifically to boot disk problems. (See the OS9 Newsletter, Volume III Issue 11B, November 30, 1992)

You'll be needing a few tools handy to follow along with this article and if your planning to just practice for the eventuality of a crashed disk then for God sakes use a backup of something.

-----------------------------Tool List----------------------------

OS9 Level II Manual, the big thick one that comes with the OS9 Level II disks.

DED -- Copyright 1987 by Doug DeMartinis -- preferably the edition below

Header for:	dEd		This edition has a patch made to it
Module size:	$17A2	#6050	so that it will recognize and
Module CRC:	$299A3F	(Good)	identify the Bit Allocation Map or
Edr parity:	$8F		BAM for short. The BAM is also
Exec. off:	$0665	#1637	sometimes called DAM or Disk
Data Size:	$0316	#790	Allocation Map....
Edition:	$05	#5	
Ty/La At/Rw:	$11 $82		Besides its the one I use. You could
Prog mod, 6809 obj, re-en, R/O			also use Qtip, but displays will differ

A screen dump utility of some sort would also come in handy for this exercise in avoiding futility. See the end of this article for suggested files and source(s).

Now to dive in there and rescue Data <Grin>.........

The first thing will cover is the breakdown of Logical Sector Number zero on any OS9 disk, as well as the invocation of DED. All numbers with a preceding $ are in hexadecimal (base 16) and others will be in decimal (base 10).

To invoke DED for this exercise type the following from any OS9 prompt on any 80x24 or 80x25 text screen or graphics screen with stdfonts merged.

OS9: DED /Dx@   (where x = the drive number with the disk to be worked on in it.)

Note: The @ in the above command allows us to open any disk just as if it were a file by itself, thus allowing us to work with any and all data the disk contains with the exception of high density disks in most cases.

Here is an example of my LSN $00. Offsets (Relative Addresses) are read as LSN+left most column+top row. See the ** in the last row? This would be read as LSN or $00+column or $70+top row or $04 or a total offset of $0074. If we were on sector 1 then it would be $0174.

The definitions of the important bytes follow the excerpt. Also see page 5-2 in the technical reference section of the OS9 Level II manual.

LSN=$00   00

```
     0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F    0 2 4 6 8 A C E
00:  00 0B D0 12 01 7A 00 01 00 00 03 00 00 FF D4 E3   ..P..z....... Tc
10:  07 00 12 00 00 00 00 00 00 00 5D 03 08 10 2D 52   ..........]...-R
20:  69 42 42 53 20 43 6F 6D 6D 61 6E 64 73 20 44 69   iBBS Commands Di
30:  73 EB 00 00 00 00 00 00 00 00 00 00 00 00 00 01   sk..............
40:  01 03 21 03 00 54 02 00 00 12 00 12 03 09 00 61   ..!..T.........a
50:  40 00 00 00 00 00 00 00 00 00 00 00 00 74 2D 00   @............t-.
60:  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
70:  00 00 00 00 ** 00 00 00 00 00 00 00 00 00 00 00   ................
```

Name	Relative Address	Size (Bytes)	Use or Function	Mine
DD.TOT	$00	3	Number of sectors on disk.	$000BD0

DD.TKS	$03	1	Track size in sectors.	$12	
DD.MAP	$04	2	Number of bytes in allocation map.	$017A	
DD.BIT	$06	2	Number of sectors per cluster	$0001	
DD.DIR	$08	3	Starting Sector of of Root Dir	$000003	
DD.OWN	$0B	2	Owners ID number (usually 0)	$0000	
DD.ATT	$0D	1	Disk attributes	$FF	
DD.DSK	$0E	2	Disk identification (internal use)	$D4E3	
DD.FMT	$10	1	Disk Format, bit mapped	$07	
DD.SPT	$11	2	Number of sectors per track	$0012	
DD.RES	$13	2	Reserved for future use	$00	
DD.BT	$15	3	Starting sector of bootstrap file	$000000	
DD.BSZ	$18	2	Size of bootstrap file	$0000	
DD.DAT	$1A	5	Date of creation (Y:M:D:H:M)	$5D0308102D	
DD.NAM	$1F	32	Disk name, last char has MSB set	see above	
DD.OPT	$3F		Path descriptor options		

Probably the most important byte to us here are the bytes at offsets $08, $09, and $0A which tell us where the root directory begins. Speaking of which, that is our next stop in the CoCo Zone....

LSN=$03 03

```
     0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F    0 2 4 6 8 A C E
00:  BF 00 00 5D 04 1A 0D 0A 02 00 00 01 20 00 00 00   ?..]........ ...
10:  00 00 04 00 07 00 00 00 00 00 00 00 00 00 00 00   ................
20:  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
30:  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
40:  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   .................
50:  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
60:  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
70:  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
80:  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
90:  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
A0:  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
B0:  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
C0:  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
D0:  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
E0:  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
F0:  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
```

Well here we are exactly where LSN $00 said the root directory starts, at LSN $000003. But where are the filenames, you ask? Well they start on the next sector.

This sector is called a File Descriptor sector or FD for short. Every file or directory on an OS9 disk has one of these. This is why you can't store a true 360K worth of files and user data on a DSDD 40 track drive for example.

For our purpose I'm going to skip the explanation of the first 16 bytes and get on with what we need from this sector to start finding data.

Starting with offset $10 ($0310) is what is called a segment list. This segment list tells OS9 where a file or directory on disk is located and how many sectors that file or directory occupies. There are 48 of these segments available each being 5 bytes wide. For you programmers, think of it as a two dimensional array such as: DIM segment(48,5). What this means is that your file or directory can occupy space in 48 different locations on disk if it is badly fragmented.

In this case mine only occupies one segment starting at LSN $0400 and is 7 sectors in size. So guess where our trip through the OS9 disk takes us next? If you said sector 4 or offset $0400 your right!

LSN=$04 04

```
     0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F    0 2 4 6 8 A C E
00:  2E AE 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
10:  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 03   ................
20:  AE 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
30:  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 03   ................
40:  43 4D 44 D3 00 00 00 00 00 00 00 00 00 00 00 00   CMDS............
50:  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 0B   ................
60:  53 59 D3 00 00 00 00 00 00 00 00 00 00 00 00 00   SYS.............
70:  00 00 00 00 00 00 00 00 00 00 00 00 00 00 07 0A   ................
80:  72 69 62 62 73 2E 63 66 E7 00 00 00 00 00 00 00   ribbs.cfg.......
90:  00 00 00 00 00 00 00 00 00 00 00 00 00 00 07 2F   ............... /
A0:  72 69 62 62 73 67 EF 00 00 00 00 00 00 00 00 00   ribbsgo.........
B0:  00 00 00 00 00 00 00 00 00 00 00 00 00 00 02 AC   ...............,
C0:  4D 45 4E 55 D3 00 00 00 00 00 00 00 00 00 00 00   MENUS...........
D0:  00 00 00 00 00 00 00 00 00 00 00 00 00 00 07 42   ...............B
E0:  4C 4F 47 D3 00 00 00 00 00 00 00 00 00 00 00 00   LOGS............
F0:  00 00 00 00 00 00 00 00 00 00 00 00 00 00 09 A3   ...............#
```

Well here we are, finally. The root directory. Again for our purposes I'm going to skip the first 32

bytes of this sector.

Each entry for each file or directory is composed of 32 bytes. 29 of them represent the file or directory name while the last 3 tell where to find that individual files or directories FD is located on the disk. Looking at this perhaps you can see the importance of having your directory names in ALL UPPERCASE and your file names in all lowercase.

In this example I have 4 directories (CMDS, SYS, MENUS, and LOGS) and two files (ribbs.cfg and ribbsgo). Lets start with a file, ribbsgo in this case.

Its entry starts at offset $A0 ($04A0) and ends at $BF ($04BF). The first 29 bytes as I said are for the file name, the last character of which has its Most Significant Bit set to mark the end of the file name. The last 3 bytes tell us where to find the FD for ribbsgo which is $0002AC or $02AC since the Most Significant Byte is 0.

So this is where we are off to next, sector $02AC.

LSN=$2AC   684

```
      0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F    0 2 4 6 8 A C E
00:  0B 00 00 5D 04 19 0C 11 01 00 00 04 A5 5D 04 19    ...].........$]..
10:  00 02 AD 00 05 00 00 00 00 00 00 00 00 00 00 00    ..-.............
20:  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00    ................
30:  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00    ................
40:  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00    ................
50:  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00    ................
60:  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00    ................
70:  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00    ................
80:  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00    ................
90:  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00    ................
A0:  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00    ................
B0:  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00    ................
C0:  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00    ................
D0:  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00    ................
E0:  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00    ................
F0:  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00    ................
```

Ok here we are. This FD is very similar to the one we examined on our way to the root directory. It contains all the same information and takes on exactly the same format as the FD for the root directory except that this time we are talking about a file and not a directory.

It tells us that our file, ribbsgo, begins at sector $0002AD or $02AD and occupies 5 sectors. So that is where we will go next. For the purposes I will only include the first and last sectors in this text as examples. I forgot to mention that we have proceeded through pages 5-3, 5-4, and most of 5-5 of the technical reference section at this point.

LSN=$2AD   685

```
      0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F    0 2 4 6 8 A C E
00:  6F 6E 65 72 72 20 67 6F 74 6F 20 66 61 74 61 6C    onerr goto fatal
10:  65 72 72 6F 72 0D 63 64 20 2F 64 64 0D 63 78 20    error.cd /dd.cx
20:  2F 64 64 2F 63 6D 64 73 0D 76 61 72 2E 30 3D 22    /dd/cmds.var.0="
30:  22 0D 64 69 73 70 6C 61 79 20 30 43 20 30 32 20    ".display 0C 02
40:  33 34 20 32 32 20 31 42 20 33 32 20 30 33 20 30    34 22 1B 32 03 0
50:  35 20 32 30 0D 65 63 68 6F 20 50 6C 65 61 73 65    5 20.echo Please
60:  20 49 6E 73 65 72 74 20 79 6F 75 72 20 4F 53 39     Insert your OS9
70:  20 42 6F 6F 74 20 44 69 73 6B 20 69 6E 20 2F 44     Boot Disk in /D
80:  30 2E 0D 64 69 73 70 6C 61 79 20 30 32 20 34 45    0..display 02 4E
90:  20 32 45 20 31 42 20 32 32 20 30 31 20 31 41 20     2E 1B 22 01 1A
A0:  31 30 20 31 39 20 30 33 20 30 30 20 30 31 20 30    10 19 03 00 01 0
B0:  31 20 30 32 20 32 36 20 32 32 0D 65 63 68 6F 20    1 02 26 22.echo
C0:  50 72 65 73 73 20 41 6E 79 20 4B 65 79 0D 76 61    Press Any Key.va
D0:  72 2E 30 0D 2A 6E 6F 6B 65 79 70 72 65 73 73 0D    r.0.*nokeypress.
E0:  69 66 20 25 30 3D 22 22 0D 67 6F 74 6F 20 6E 6F    if %0="".goto no
F0:  6B 65 79 70 72 65 73 73 0D 65 6E 64 69 66 0D 64    keypress.endif.d
```

Well we made it! The actual file data. There are no special codes or anything of that nature here to explain. Just the ASCII codes for the contents of the ribbsgo script file. With program modules it would be the hexadecial representations of the commands and variables and such within the program.

As I said there are 5 consecutive sectors (or 1 segment) that this file occupies but I will only include this and the last sector, because everything in between is technically the same.

LSN=$2B1   689

```
      0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F    0 2 4 6 8 A C E
00:  6F 63 6D 64 73 0D 67 6F 74 6F 20 2B 65 78 69 74    ocmds.goto +exit
10:  0D 2A 66 61 74 61 6C 65 72 72 6F 72 0D 65 63 68    .*fatalerror.ech
20:  6F 20 45 72 72 6F 72 20 25 2A 20 69 6E 20 52 69    o Error %* in Ri
```

```
30:  42 42 53 47 6F 2E 20 20 46 69 78 20 61 6E 64 20   BBSGo.  Fix and
40:  74 72 79 20 61 67 61 69 6E 0D 67 6F 74 6F 20 2B   try again.goto +
50:  65 78 69 74 0D 2A 66 69 6E 69 73 68 75 70 0D 64   exit.*finishup.d
60:  69 73 70 6C 61 79 20 31 62 20 32 33 0D 72 69 62   isplay 1b 23.rib
70:  62 73 6D 61 69 6E 20 23 31 36 4B 20 3C 3E 3E 3E   bsmain #16K <>>>
80:  2F 77 37 26 0D 2A 65 78 69 74 0D 64 69 73 70 6C   /w7&.*exit.displ
90:  61 79 20 31 62 20 33 32 20 30 30 20 30 35 20 32   ay 1b 32 00 05 2
A0:  31 20 30 43 0D 00 00 00 00 00 00 00 00 00 00 00   1 0C............
B0:  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
C0:  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
D0:  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
E0:  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
F0:  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
```

Ok, this is the last sector of the ribbsgo file. The important thing to mention here is that this file does not contain essential information throughout the entire sector. The file ends with a carriage return ($0D) at offset $A4 or $355 taking into account for the sector we are on.

See all the $00's following that carriage return? We don't need them. I'll explain how to get rid of them later, for now its enough for you to know that there not needed. In some cases those extra bytes may contain $E5's which is the value that OS9 writes to each sector when you format a disk.

Now that you have a basic understanding of how an OS9 disk is put together to become an effective storage medium we can go on and discuss how we are going to go about recovering data.

```
    Up/Down Arrows  Read & display Next/Previous sector
   <CR> Clean up the screen display
     *   Restart
     $   Fork a SHELL (Ctrl-BREAK to return)
*    A   Append displayed sector to output file
*    C   Close output file
     D   Diddle (adjust) file length
     E   Edit the displayed sector
     F   Find a byte or text string (BREAK aborts)
     H   Help screen (also use '?')
     L   Link to a module - List all modules
*    N   Next occurrence of byte(s) or string (Find)
*    O   Open a file for output (use with Append)
     P   Push current sector onto stack
     Q   Quit dEd - Exit to OS9
     R   Remove and display a sector from stack
*    S   Skip to given sector (sector # in hex)
     U   Unlink from module
     V   Verify all modules in file
     W   Write the sector back to the disk
     X   eXpert mode toggle on/off
     Z   Zap (fill in) the sector displayed
```

What you see above is the built in help from DED. The options we will be using most often are the starred options above.

*   S   Skip to given sector (sector # in hex)

This option will let us skip to the sector(s) that we have identified from the file descriptors (FD's) and will speed things up considerably.

*   O   Open a file for output (use with Append)

Once we have found the first sector of the data we wish to recover we can use this option to open a path to another disk (or RAM disk) on which we will store the recovered data. Since we will have to do some editing on the recovered file a RAM disk is recommended.

*   A   Append displayed sector to output file

Once we have opened the destination file for the data we are trying to recover this option will let us add the current sector to that new file. You use this until you either reach the end of that particular segment (Another FD will most likely be displayed at the end of a segment or file) or the end of the file.

```
    Up/Down Arrows  Read & display Next/Previous sector
   <CR> Clean up the screen display
     *   Restart
     $   Fork a SHELL (Ctrl-BREAK to return)
*    A   Append displayed sector to output file
*    C   Close output file
     D   Diddle (adjust) file length
     E   Edit the displayed sector
*    F   Find a byte or text string (BREAK aborts)
     H   Help screen (also use '?')
     L   Link to a module - List all modules
```

```
*  N   Next occurrence of byte(s) or string (Find)
*  O   Open a file for output (use with Append)
   P   Push current sector onto stack
   Q   Quit dEd - Exit to OS9
   R   Remove and display a sector from stack
*  S   Skip to given sector (sector # in hex)
   U   Unlink from module
   V   Verify all modules in file
   W   Write the sector back to the disk
   X   eXpert mode toggle on/off
   Z   Zap (fill in) the sector displayed

*  C   Close output file
```

Now that we have recovered the data or file we must close the file before doing anything else with it.

```
*  F   Find a byte or text string (BREAK aborts)
*  N   Next occurrence of byte(s) or string (Find)
```

If you know specific words or byte sequences to look for within the data or file your trying to recover then these two are handy for locating those words or sequences.

Well we've recovered a file or data. There is, if you recall, quite likely some extra unwanted bytes. What do we do to get rid of them? That's easy, again using DED (and ident for program modules) we diddle with the file length. Now you won't be dealing with real sector numbers, just the relative sector offset from the beginning of the file. In this case it will read LSN $00 through $04 although we may not actually be on sectors 0-4.

At any rate you need to find the last relative sector of the file probably using the arrow keys to scroll through it. When you reach the last sector look and the LSN, left column number, and top row number and determine the offset for the last byte (the carriage return) and add 1. In this example that last byte will be at $04A4 then add 1 giving us $04A5.

Hit D for Diddle with file length. It will tell you the old length and ask for the new length. Type it in ($04A5 for this example) and press enter. You will see the extra bytes disappear in front of you.

Now hit Q to quit and answer 'Y' and you have just recovered your first file. Give yourself a pat on the back, get a cup of coffee and dig in cause your gonna be dancin' on the keyboard for several hours to completely recover one DSDD disk.

It took me roughly 24 hours to recover all data from 3-3 1/2" 756K floppies (I have mine formatted for 84 tracks double side rather than the usual 80 tracks double side <Evil Grin>).

For some disks your directories will get trashed and there is little one can do to recover the directories (that I know of) in which case you will have to sit there with the arrow keys in DED identifying FD's and locating the 'lost' files. This is what took me so long, my directories got trashed.

This is, as I said, a time consuming method but I know of no program that will do it for you. If I ever get some of my other programming projects finished I intend to write something, but for now this method will have to do.

Good luck recovering that lost data!

JWT Enterprises

Optimize Utility Set Pac:

➡ Optimize your floppies and hard drives quickly and easily! ➡ Includes utility to check file and directory fragmentation.
➡ Works alone or with Burke & Burke *repack* utility. ➡ One stop optimization for any Level 2 OS-9 system.
➡ Check and correct any disk's file and directory structures without any technical mumbo-jumbo before optimizing.
➡ Run periodically to maintain the integrity of your disks as well as the reliability of your data.
$19.95; Foreign Postage, add $4.00

UpTime Back-Issues:	From September 1992 to August 1994 monthly. Limited supplies. *$1.00 each; Foreign Orders, add $1.00 each*
UpTime Volume 1 Pack:	Contains September 1992 to August 1993 issues in binder. Limited supplies. *$12.00 each; Foreign Orders not available.*

Nine-Times Back Issues:
From May 1989 to March 1994, bi-monthly intervals
• Helpful and useful programs
• C and Basic09 programming examples
• Hints, Help columns, and informative articles
• All graphic/joystick interface
• Can be used with a hard disk or ram disk
• **Write for back issue contents.**
$4.00 each; Foreign Orders, add $1.50 each
Bulk orders of six or more, $3.00 each; Foreign Orders, add $1.50 each
Entire set of 30 issues, $60.00 ($2.00/issue); Foreign Orders, add $35.00

Magazine Manual:
Instruction manual for the *Nine-Times* disk magazine.
• Includes setup and booting information
• Originally sent to all new *Nine-Times* subscribers
$1.50; Foreign Orders, add $2.00

Magazine Source:
Full Basic09 code and documentation for the presentation shell used with *Nine-Times*.
$14.95; Foreign Orders, add $5.00

Foreign postage excludes U.S. Territories and Canada. These products for OS-9 Level 2 on the CoCo 3. Sorry, no C.O.D.'s or credit cards; Foreign & Canadian orders, *please* use U.S. money orders. U.S. checks, allow 4 weeks for receipt of order. Ohio residents, please add 6% sales tax.

Glenside to Sponsor Fourth Annual "Last " Chicago CoCoFest!

Here are the 5 "W's"

WHO?   The Glenside Color Computer Club of Illinois presents

WHAT?   The Fourth Annual "Last" Chicago CoCoFEST!

WHEN?   April 29th & 30th, 1995 (Sat. 10am-6pm; Sun. 10am-4pm)

WHERE?   HOLIDAY INN ELGIN (A Holidome Indoor Recreation Center)
          345 W. River Road (A city block from I-90 & IL-31S)
          Elgin, Illinois   (Same great location as last year!)
          Overnight room rate: $55.00 (plus 10% tax)
          Call 1-708-695-5000 for reservations.
          Be sure to ask for the "Glenside" or "CoCoFEST!" rate.

>>>>> YOU MUST REGISTER UNDER COCOFEST TO GET THIS RATE <<<<<

WHY?   A. To provide vendor support to the CoCo Community
          B. To provide Community support to the CoCo Vendors
          C. To provide educational support to new users.
          D. TO HAVE AN OUTRAGEOUSLY GOOD TIME!!!!!

And now, the "H" word.

HOW MUCH?   Admission:
             $8.00 at the door, 2-day pass
             $5.00 at the door, 1-day pass

Advance ticket sales:
             $5.00 + SASE for the whole show or
             $5.00 + $.50 postage & handling
             Children 8 and under   FREE

Contact: George Schneeweiss, Treasurer
Glenside Color Computer Club
RR#2   Box 67
Forrest, IL 61741-9629

For further information, general or exhibitor, contact:
Tony Podraza, Fest Chairman
708-428-3576, VOICE
708-428-0436, BBS

Meeting Location:

Glenside Public Library
25 West Fullerton Avenue
Glendale Heights, Illinois 60172

Directions:
The Library is on Fullerton Avenue, about one-half mile West of Bloomindale Road. *Fullerton is about one mile South of Army Trail Road and about one mile North of North Avenue (Route 64).* Bloomingdale Road is about two miles West of Route 53/I-355/the North-South Tollway.

Next Meeting Date
April 13, 1995

Glenside Color Computer Club
31 South Edgewood Avenue
Lombard, IL 60148