****

THE AWAUG NEWSLETTER

****

ADAM Washington Area User's Group

Vol. 7, No. 3

June 1991

In This Issue

Featured this month is Ron Collins' second article on CP/M starting on page 7. I have published the entire 12 and 1/2 pages because it's just too good to break up into small pieces. There's a lot of great info there. Don't be put off by the title....Advanced CP/M. There's lots of good info there for everyone. From ANN, we have an excellent variety of articles starting with a super writeup of his great program UNDELETE.COM by Guy Cousineau starting on page 19. Dean Roades provides an update of ADAMCON 03 right here on page 1. Norm Castro provides some info on the early Colecovision magazines on page 2. Dan Ryan shares a few of experiences with Adam repairs on page 5. Barry Wilson lets us all in on the secrets of ANN on page 3 and provides an update on the **ADAM SURVIVAL GUIDE** on page 23. The astute observer will notice a change in this months newsletter. In an effort to hold down costs, I've switched to a 12 pitch font from 10 pitch and tightened the margins up a bit. We wind up with more words per page. The goal was to try and keep the newsletter under 20 pages, but there was just too much I wanted to include. Tom Barrett and the Postal Service (now there's a real oxymoron) will just have to cope. Have a great summer! JM

From the 5-91 463 ADAM Newsletter

ADAMCON 03 UPDATE - 4/20/91
by Dean Roades
Convention Chairman

Things are really rolling. I look forward to going to the mailbox each day because there are usually new registrations. ADAM suppliers are working feverishly on new software and hardware to unveil. Plans are falling into place and people are preparing information for the workshops at AC03.

This month, I want to address the people I refer to as "BASE ADAMites". They are known by many names, NewBees, novices, beginners, but they all have one thing in common - ADAM. They use their ADAM the way it came out of the box, maybe adding a second tape drive or a disk drive. They enjoy ADAM for all the things it can do for them.

Dear BASE ADAMites,

If there is any way you can get to South Bend in August for the convention, GO! ADAMCON 03 is geared for the novice ADAM user. It will be the BEST place to get answers to your questions, solutions to your problems, and give you a unique opportunity - buy ADAM software, hardware and supplies off the shelf!

We have assembled an excellent cross section of ADAM vendors who will display their wares. Ribbons, data packs, printer wheels. All the basic supplies you need will be right in front of you.

We are also fortunate to have the best minds in the ADAM world making plans to be there. I honestly don't think there is any question about ADAM that can't be answered by these people. They are the people whose articles you have been reading in your favorite newsletter.

The best part of the first two conventions was the people. When I attended ADAMCON 01, I was a little apprehensive. After all, I was basically a novice with no special skills in any particular area of ADAM. When I arrived, I immediately felt comfortable. I met people for the first time and soon felt like I had known them all my life. Everyone was eager to help and willing to spend the time to answer my many questions. ADAMites are more eager to talk about ADAM than a grandparent about their grandchildren.

The point I am trying to make is that we would like to see you at the convention. You may not know anyone when you arrive, but you will when you leave. Set the time aside now and make plans to attend.

We are looking forward to meeting you.

****

COLECOVISION EXPERIENCE MAGAZINE

Coleco published something called the Colecovision Experience Magazine in three issues.

Vol. 1 Number 1 Spring 1983
Vol. 1 Number 2 Fall 1983
Vol. 1 Number 3 Winter 1984

These are available from Norman Castro 809 W. 33rd Ave. Bellevue NE 68005 at $3.00 each Post Paid or all Three for $8.00 Post Paid. {Foreign all three issues for $10}

Assorted Information, hints, etc. from these publications includes:

GAME TIPS:

SPACE PANIC: You can save oxygen by chasing after the creatures on screens 1-4. ON screens 5 & higher try to get rid of all the red creatures first. TWhey are the easiest and leave you more time to concentrate on the Bosses and Dons.

PEPPER II: Complete one maze at a time so you wont have to back track later. Enclose the room by quadrants and always remember to energize first so you can freeze the Zipper Ripper and gobble up the roaming eyes for bonus points. Close off a corner room last so ou dont have to cross in front of the eyes again.

COSMIC AVENGER: Get down at level of tanks and shoot them instead of bombing them. OR save the tanks for later and get the UFOs first because tank fire can hit the UROs and UFOs are more of a threat. The Yellow Missle housings are bad because those missiles track you and will get you if you dont move fast. Try flying low and keep the fire buttons pressed all the time.

SPACE FURY: No reason to thrust on the 1st screen; just concentrate on getting rid of the alien ships. Given the choice of three motherships, dock with the top one first. Dock with the right one second and the left one third. Eaach mothership gives different amounts of firepower and lets you shoot in different directions, they above order gives the firepower needed to destroy the aliens quickly. Always shoot out at least one of the group of four small enemy ships as it takes four small ships to form one large enemy ship.

DONKEY KONG JR.: 1st screen: use the fruit to knock off the Snap-Jaws, wait until just below you. 2nd Screen: start climbing the third and fourth chains and work over to the right side of the screen. Then go back to the 1st two chains. Best way to clear the screen and get points. 4th Screen: As you hit the jump board press the jump botton and be bounced up to the hanging vine.

LOOPING: Dont fly too close to the ground in the city because the ballons may catch you by suprise. Dont use the accelerator in the maze. Try to shoot the green drop before it leaves the spigot. Hold the firebutton down.

ZAXXON: When reaching open space for the 1st two groups of three fighters stay at the bottom right of the screen and move in a horizontal position while firing. For the remaining enemy fighters, stay at the top of the scfeen while firing. Best way to get thru the force field is to stay on the bottom, then come up slowly to line up your tire/fire ? with the opening in the force field.

VIDEO-CARTRIDGE LIFE: Cartridges have NO moving parts so they cannot wear out that way. The only thing resembling aging in a game cartridge is very slight wear on the connecting boards which occurs every time you insert or withdraw the cartridge. Coleco engineered the cartridges to withstand up to 10,000 such insertions, which would be putting a cartridge in your Colecovision three times a day each day for almost ten years.

As for breaking the bank by scoring too high, if you go over the capacity of the screen's digits to show the high score, it automatically rolls over to zero. (Gee thats a problem I often have. :))
LADY BUG: There are four selectable skill levels with 18 progressively more difficult boards in each. The last vegetable is a horseradish.

Other such info is contained in these.

An interesting view of the early publications of ColecoVision.

****

FOR THOSE THAT WONDERED.....

Gee, I enjoy getting my ANN disks each month and reading all the information on them but I have wondered, how does the information get to ANN and to me.

Well in response to this much asked question the Official ANN response as stated in the Policy Manual under Article 2 Section 12 paragraph 3 subpart A7 is as follows:

The stork brings them. Further you should not ask such questions until you are older, go to bed without dinner.

Now for those of you willing to take the unofficial version for which I make no guarantees, warranties, or other such creatures of law and which I will fully deny under moderate torture, such as being forced to listen to any of the following: (1) Ron Collin's The History of CPM thru the Middle Ages; (2) Rob Friedman's CPM as a second language for the foreign born; (3) Tom Keene's The Care and Feeding of a CPMer, and its sequel, Raising CPMers for Fun And Profit, here it is.....

There are hidden Adamite moles, who many years ago we planted in various cities to further the interests of ANN. Now when they are needed, they come to the surface (hence the name moles) and then they send threatening letters to local adamites strongly suggesting that they see that their medical and accident insurance is all paid up as they will be using it shortly unless they submit articles to ANN. If this fails, well we are not in this business for this long without having the necessary contacts to see that pressure is applied to wrists, kness & elbows. I would suggest investing in Plaster Cast Co. stock as it may be a growth industry.

The articles are then sent in plain brown envelopes without return address to ANN Central which is Barry Wilson St.Louis Mo. USA.

Here Barry places them unread on disks (it helps keep the sense of mystery alive not to know what is in them).
When the supply of information sent is not sufficent (like every month) Barry must use his skill and daring to obtain more material.

Sometimes this is by signing on various BBSs under assumed names and stealing (downloading) items meant for

downloading and some meant only for private communications. Sometimes it is necessary to go to the archieves and look through old dusty disks from years ago, to find items to pull off and put on current ANN disks, hoping all who read the original many long years ago have now gone on to the Great Coleco Co. in the sky.

Sometimes it is necessary to write nonsense items such as this item, merely to fill out the space and not necessarily to impart knowledge. Are these items works of fiction, friction, fixation, or fact. Only the Shadow knows.

As each disk is filled, except for a small space for the description (readmetwo) file, it is sent by truck to Florida, then by boat to Cuba, where it is flown to the South Pole, dog sleded to a secret military base wehre it is taken by submarine to the North Pole and finally delivered by reindeer or by an elf to Dean Roades in Indiana. (ANN Central-2, which is a duplicate of the ANN Central-1 and meant to serve as the main control centre for ANN in case of an emergency, attack by CPMers, or other natural or unnatural disaster.)

Here in Indiana at ANN Central-2, Dean formally hands the disks over to Joe Alford in a ceremony suitable for photo taking, known as the changing of the disks. However, prior to this very colorful and historical and hysterical ceremony, Dean makes certain additions to the disks. These are trade secrets, handed down from the time of Big John and Sol Swift thru generations of adamites until Dean's great great great grandfather who bequeath the same to Dean.

Joe Alford who has aged considerably under the strain and responsibility but who being a loyal adamite continues the good fight waged by ANN against evil, ignorance and CPM has a special and very difficult job to do.

To Joe Alford serving in the finest tradition of such greats as Rich Clee, Ron Mitchell, David Cobley, Pat Herrington and others of similar stature, goes the thankless, unrewarding, unrelenting and just generally boring job of ACTUALLY READING EACH AND EVERY FILE. This is done to enable Joe to write the Readmetwo, item descriptions, file. What it amounts to is many days of work to come up with a condensed, organized and detailed while being brief and to the point, file of just a few K to let you know what is on each disk.

Now after all of this, each disk must be cleared with the National Security Agency for Canada , United States, Great Britian, as well as other countries whose names are classified and therefore cannot be given out at this time. (Noticed any Russian or German letters in your files ???)

Joe reads each file and tries to determine the often hidden and secret meaning and subject of each file. Then locking himself in his room, with food & drink, much drink for a 3-4 days stay, does his magic and produces the Readmetwo file.

The disk is then returned by a secret, long, and twisted route, again to conceal the source and each person's involvement in this project, to the North Pole where it is again reindeered into Dean Roades in Indiana.

The master disk is now laboriously copied by hand by a team of Monks from the Carmelotta Monastery, a well known but ancient order, sworn to maintain silence and ANN & Adam secrets. Since the assemblage of so many Monks in one place might cause notice and suspicion the disk copying actually takes place in Indiana, Washington State, Missouri and Florida. Nuns are used to get the disks from ANN Central-2 to the various copying locations.

At each copying location, Monks using double sided styluses or is it stylusi or whatever they use to write on parchment and upon plastic, make the necessary copies, properly label them as supervised by the Food and Drug Administration.

The reader will be relieved to know that ANN disks are cholesterol free, Sodium free, non-caloric and can be used to pick your teeth for good dental hygiene.

But I digress as is my custom when I am thinking of what to next write down for your enlightenment (?).

After the Food & Drug Administration supervises the proper labeling of the disks, they are then placed into sterilized disk folds, placed in appropriate mailing containers, weighed, measured, tested for flavor and then the final step, one which has much ceremony almost as formal as the changing of the disks above mentioned, is the applying of the stamps and postage.

Now your ANN disk is ready for you and when you look in your mail box and find it, you can be sure that.....

The stork brought it.

Your Editor, Barry Wilson.

For those foolish enough to do so, full permission is given for reprinting this file.

Further, those with opposing views are invited to apply sufficent lubricant to this disk and then insert in the proper orifice.

****

Some Thoughts on Repairs
by Dan Ryan

Repairs are the subject of a lot of ADAM talk. Where to get them, how long they take, how much they cost, and will I really EVER get my equipment back if I send it to this stranger who says he does ADAM repairs. A good topic of conversation.

I have had a few ADAM repairs done since 1986 and believe I know the general ropes in this area. I haven't used every repair person in the known ADAM world, and I believe most of those I haven't used are VERY reputable and competent ADAM repair folks. I can only speak from personal experience, which has all been GOOD!

My ADAM repair history dates back to December 1986 when I pulled my new ADAM, purchased for $200.00, out of the box. What a thrill, but what a chill to discover immediately that the reset switch did NOTHING. Honeywell did reapirs in Atlanta on ADAM back then and they turned it around under warranty in about two weeks. A couple of months later the problem reappeared and they fixed it again, still under that six month warranty.

The next time I had ADAM trouble, Coleco was in Chapter 11 and Honeywell had nothing to do with the ADAM anymore. I had purchased software from Terry Fowler and subscribed to hisuccess. Several other repairs have been performed by Terry for me since then and all have pleased me, including a disk drive repair in late 1988. That's ADAM'S HOUSE, of course. I promise you won't go wrong by doing business with Terry!

I don't know if Leroy McKenzie still works on ADAM in Ope Locka, FL, but he's done about four ADAM repairs for me (console) all with complete satisfaction.

Of late I'm impressed with price and performance from Dan Elliott. Dan works on most anything common to ADAM, including the 64K card and disk drives. He repaired a disk drive for me in about six weeks for under thirty bucks. Since he advertised four to six week turnaround, I was completely pleased with the service. Another local ADAMITE, referred to Dan by me, reported a just over ONE WEEK turnaround of his disk drive. That friend thinks I'm quite an ADAM advisor now. Another local user sent his ADAM tape drive to Dan (yes, he repairs the data drive) and it appears the completed repair will amount to just under thirty dollars. I'll tell you folks, Dan has the best prices in the business. That's:

Dan Elliott
Rt.1 Box 117
Cabool,MO 65689

By the way, Dan, this plug entitles me to one free repair, not to exceed $1,000.00 in value!! (smile!)

Just some thoughts on ADAM repair, since if you are not a veteran of the ADAM repair circuit, it's hard to decide where to send your equipment. The problem with writing this article is that I realized how many repairs I have had done over the years. You can bet I have a couple of spare ADAMS tucked away in the closet!!

****

AWAUG MEMBERSHIP INFORMATION

AWAUG is an organization dedicated to the support of the ADAM computer system and the furthering of computer literacy. Membership in AWAUG costs $15 per year. Members receive a bi-monthly newsletter and are entitled to purchase items offered by the club such as public domain software, WordStar 3.0, formatted tapes, and some hardware. Our volunteers can help to answer your questions.

Meetings are held on the third Saturday of each month at the Tyson's Pimmit Regional Library at 11 AM. The address is 7584 Leesburg Pike, Falls Church, VA (1/2 mile inside the beltway on Route 7).

Meeting are free. For more information about the meetings or the club, contact Jim Howard at (703) 960-5315 or check in on the AWAUG BBS at (202) 561-2475 (2400/1200/300 8-N-1).

Your club officers are:

Jim Howard	President & Librarian
Scott Gordon	Vice President
Ralph Mason	Treasurer
Jeff Jodoin	BBS SYSOP
Tom Barrett	Membership
Jack MacKenn	Editor
Mark Gordon	Meeting Dir
Cliff Sinopoli	Procurements

****

ADVERTISE

IN THE AWAUG

NEWSLETTER!

The AWAUG newsletter accepts advertising for publication. We believe that advertisements help ADAM owners to know what is available and, therefore, that they are a good service. In keeping with this notion, our rates are very competitive. Because it is to everyone's advantage to encourage enrollment in our club, we offer a discount to members who wish to advertise.

At present, our membership stands at just under 100 ADAM owners, so your advertising will reach a significant number of very interested people. Rates are as follows:

SIZE	Non-member	Member
Full page	$9.50	$6.35
Half page	$6.00	$4.00
Quarter page	$3.00	$2.00
Column inch	$1.00	$ .65

A column inch is 38 characters wide and 5 lines high. For each consecutive column inch, add one line.

Advertising may be for any legal goods or services. Material may be submitted in any intelligible form, from camera ready artwork to telephone orders voice, or by data transmission via the club's BBS, telephone (202) 561-2475.

To make inquiries or place an ad, call or write:

Jack MacKenn (703) 371-7548
415 Camden Drive
Falmouth, VA 22405

****

ADVANCED CP/M

By Ron Collins

INTRODUCTION:

For part two of my CP/M articles, James Poulin asked if I would write about my patches to various CP/M software. What he wanted was a guide of some sort to explain when and how to patch program and overlay files to turn them into ADAM compatible programs. I have, by trial and error, been able to install a wide variety of software on my own home ADAM system. It is my hope that what you read here will show how really easy it is to modify your software all by yourself.

The common feeling among those that use CP/M (and the publicity I remember advertising it tended to support this) was that any CP/M compatible computer could use any program written in CP/M. When it was first released to the market in the early 1970s, CP/M was advanced as an long awaited answer to the delima faced by home computer buyers. At that time, each computer had its' OWN main operating system. There was NO way to use the software from any one type on any other type. With no compatibility there was also no real hope for growth. There was only competition. CP/M was to end all of that confusion by providing a common interface from one system to another. Indeed, in only 5 years from its' date of release, more than 300 different computer systems were operating under the CP/M environment!

Todays' prevelant MS-DOS operating system has carried this idea into the '90s' by doing effectively the same thing. Back in those days, however, there was only CP/M... and WHAT an only it proved to be! A program in fact COULD run on ANY CP/M computer IF the hardware supported by the program was equal to the task. Take a look at the basic ADAM computer system. It works on a TV screen with a 32 character per line display. It uses digital data drive storage media. It has a daisy wheel printer and can use a modem plugged inside it that, like the rest of the computer hardware, is NOT compatible with any other known computer system!

Each of these components is used and spoken to by the computer in a specific way. To have any one program support all of the many features of over 300 types of computers would be impossible. The size of the program would be far to large for even a hard disk to fully support. The answer to the dilemma was for each computer to have a version of CP/M as the operating system. That individual version of CP/M had all the knowledge needed to speak to each and every component piece of hardware connected to the computer. All any program then needed to do was be able to talk to CP/M. IT, the operating system, would then go out and translate the program information into something that computer could understand.

A major drawback in all of this was the use of machine specific program routines. Most of these dealt with special codes to make a program appear on the display device in a specific manner. Some codes told the system how to use inverse video. Some told it how to do a line feed or return to normal video. Some were used to do the routine needed to underline text onscreen. Some were detailed video routines used to create screen graphics displays. Some are only there to clear the screen and "home" the cursor to the top left corner of the display screen.

In addition to such things as the setting of proper screen codes, the patches needed, beyond those supplied by CP/M, involved codes for altering drive attributes, changing file names, etc. Since no two programs need exactly the same "fine tuning" to work from an ADAM, I'll try to give you a basic outline of what you will do to complete the needed patching of programs for your own ADAM.

In terms of patching, there are three classes of change a program might need. The

first class is simply one of following supplied menu selections until you have let the INSTALL.COM or equivalent program plug in the new values for you. The second class is that kind when a program is provided with an overlay file that you the user must alter for your own system needs. The third and final of the patch classes is the type of a program I have run into all too often. One that is written specifically for some other computer of the more than 300 types of computer and that uses quite different screen codes, etc. This type is the hardest of all sometimes to "fix" for an ADAM, but with a bit of patience and sticking with it, these too can be patched.

THE PRELIMINARIES:

Before we get started working with our patch programs, we should first find out what we need to change in our selected program. A quick look in the CP/M 2.2 and ASSEMBLER manual pages D23 to D31 will give you a good working start on what codes the native ADAM TV display will respond to. This screen code allocation is not a simple emulation of a well known terminal as many have thought in the past. Rather, the ADAM's TV display is a composite of the ADDS terminal, the DEC VT-52 and the Heath/Zenith H19 displays. Doing a patch for the ADAM's TV display is not always such an easy thing to do. I've found a great contradiction in what works best with different programs. Some, I have installed as ADDS displays. On others, the ADDS didn't "work right" but worked just fine when I installed them as a VT-52! Todays' ADAM can use T-DOS which emulates strongly the Heath/Zenith display. The many buyers of RS-232 serial ports connected to dumb terminals, each with its' own particular display requirements, just serve to make the problem more complex!

In each of these cases, no single display patch would give me all of the screen attributes I wanted. I generally had to go into it and plug in a few Heath screen codes to complete the job. That's why I say the ADAM's TV display is a composite or hybrid. I have assembled a list of some of the more popular screen codes that you can use as a reference. Before doing anything else, take a good look at these codes as you may be using them yourself soon!

* *

ESCAPE CODE AND CONTROL CODE TABLE
ORPHANWARE BUSINESS SYSTEMS
80 Column Video Unit

Emulation	**H-19**	**ADM-3A**	**BEEHIVE**	**ADDS 200**
CURSOR FUNCTION		***ESCAPE AND/OR CONTROL CODES***		
Carriage Return	^M	^M	^M	^M
Line Feed	^J	^J	^J	^J
Back Space	^H	^H	^H	^H
Tab	^I	^I	^I	^I
Home	Esc H	Esc H	Esc H	Esc A
Up	Esc A	Esc A	Esc A	
Down	Esc B	Esc B	Esc B	
Right	Esc C	or ^L	Esc C	^F
Left	Esc D	Esc D	Esc D	^U
Reverse LF	Esc I	Esc I		
Direct Cursor Add	Esc Y	Esc =	Esc F,Y	Esc Y
Read Cursor Adress	Esc n	Esc n	Esc G,\	
Save Cursor Posit	Esc j	Esc j		
Recall Curs Posit	Esc k	Esc K		

EDITING AND ERASING

Clear Display	Esc E	or ^Z	Esc E	^L
Erase to Top	Esc b	Esc b		
Erase to End	Esc J	Esc J	Esc J	Esc k
Erase Line	Esc l	Esc l		
Erase Left	Esc o	Esc o		
Erase Right	Esc K	Esc K	Esc K	Esc K
Insert Line	Esc L	Esc L	Esc L	Esc M
Delete Line	Esc M	Esc M	Esc M	Esc l
Insert Character				Esc F
Insert Char Mode	Esc @	Esc @	Esc Q	Esc F
Insrt Char Mod off	Esc O	Esc O	Esc @	
Delete Character	Esc N	Esc N	Esc P	Esc E

MODE CONTROL

Disable Keyboard	Esc }	or ^Q	Esc c	^Y
Enable Keyboard	Esc {	or ^N	Esc b	^B
Reverse Video	Esc p	Esc p	Esc l	
Normal Video	Esc q	Esc q	Esc m	
Graphics Mode	Esc F	Esc F	Esc R	
Exit Graphics Md	Esc G	Esc G	Esc S	
Wrap Around ON	Esc v	Esc v		
Wrap Around OFF	Esc w	Esc w		
Alt Char Set	Esc f	Esc f		
Normal Char Set	Esc g	Esc g		
Cursor on/off			Esc Z	
Next Char Graphic				Esc Z
Set Mode Control	Esc x	Esc x		
Reset Mode Cont	Esc y	Esc y		
Set Baud Rate	Esc r	Esc r	Esc 7	Esc A
Test Mode	Esc t	Esc t		
Set CRTC Mode	Esc s	Esc s		
Reset to Power on State		<--->Esc z	Esc V	Esc s

MISCELLANEOUS

Escape	^[	^[	^[	^[
Bell	^G	^G	^G	^G

* *

"What are screen codes anyway?" and "Why do I need to know them?" are probably two of the most often asked questions I get from people. Well, simply put, screen codes are ESCAPE codes (those that combine the ESCAPE key to be followed by another keyboard character to initiate an action) and CONTROL codes (those that combine the CONTROL key to be followed by another keyboard character to initiate an action). These codes, no matter what the type, tell your computer how to put things onscreen. While it may seem that just sending it to the TV is all that happens, there is really far more going on behind the scenes, so to speak.

First of all, there must be a way to erase what is already on the screen before you can put something new on it. Next, you must have a way of putting the cursor at the top left corner of that screen so that anything new will start at the screen's beginning. After that, you need a way to type a line and have the cursor move down

one line for the next typed line. Without this, you will end up typing over the previous line over and over again.

Now, what if you type all the way to the bottom of the screen? You will need a way of getting the ADAM to "scroll up" for you. That way, you will get more lines to type on than just the one screen you are looking at. You will also need codes that will let you insert or delete something should changes in your text be needed. If you wanted to make something onscreen more dominant, you would want to use inverse video on some parts of your text. This means you need a code to turn on the inverse video. You will also need a code to turn this inverse back off so that the rest of your letter isn't inversed!

If you look at the above ESCAPE code chart you will see some of the uses of these codes. Some of them will erase the screen and some of them will home the cursor to the above left corner of the screen as well as clearing the screen. There are codes that move the cursor to the right, to the left, up or down. There are some used to insert a letter or a line. There are some codes that let you highlight words or even whole sentences. If you are asked to fill in the code for a certain action when trying to install a program, the ESCAPE CODE chart will give you an idea of what you need to type in.

INSTALLING WITH "INSTALL.COM" PROGRAM

If you are lucky enough to find a file on your new program disk called INSTALL.COM (or a similar name), then you are well on your way to being "up and running" on your ADAM. Most of these files started as a set of Turbo Pascal routines. After being polished and them combined, the files have been "compiled" into a program that can be executed by simply typing its' name. Probably one of the most useful features of a Turbo Pascal program is its' easy installation facility. You guess it... INSTALL.COM!

There is even a PDTINS.COM file in the public domain. This one will let you patch just about ANY Turbo Pascal oriented program with ease. In any event, the installation is done for you with only your needing to select from a list of options. The normal procedure is it type in the name of the INSTALL program and press the RETURN key. You will be presented with:

Turbo Pascal general installation System
Choose installation item from the following:

Screen installation | Quit

Enter S or Q:

At this screen, you will want to press the "S" key which will tell INSTALL.COM that you want to do a "Screen installation."

You will then be presented with a screen that looks like this:

1) ADDS 20/25/30
2) ADDS 40/60
3) ADDS Viewpoint-1A
4) ADM 3A
5) Ampex D80
6) ANSI
7) Coleco ADAM/80 CVU
17) Otrona Attache
18) Qume
19) RC-855 (ITT)
20) Soroc 120/Apple CP/M
21) Soroc new models
22) SSM-UB3
23) Tandberg TDV 2215

8) DEC Rainbow, 8 bit
9) Hazeltine 1500
10) Hazeltine Esprit
11) Kaypro
12) Lear-Siegler ADM-20
13) Lear-Siegler ADM-31
14) Liberty
15) Morrow MDT-20
16) Osborne 1
24) Teleray series 10
25) Teletex 3000
26) Televideo 912/920/92
27) VISUAL 1050
28) Visual 200
29) Wyse WY-100/200/300
30) None of the above
31) Delete a definition

Which terminal? (Enter no. or ^Q to exit):

As you can see in the above list, there is a selection (number 7) that is listed as being for a "Coleco ADAM/80 CVU". If you were to select this one, your program would be installed for the 80 column Heath/Zenith video emulation. This is also a good choice if you are installing your program for use under T-DOS on a TV.

Just pressing a "7" will choose this one for you. If you don't have an Orphanware 80 CVU or a Heath/Zenith terminal, but are to use this program under standard CP/M using a TV display, you will want to choose one of the other selections. Always patch only a backup copy of the program. You wouldn't want to inadvertantly do anything to ruin your new program now would you?

After you make your selection (which ever one looks right to you) the screen will prompt you:

Do you want to modify this definition before installation? (Y/N)?

An answer of "Y" will give you a chance to plug in some of those values we saw listed in the ESCAPE CODE listing. An answer of "N"
will take you to the next phase of the installation:

Hardware dependent information

Operating frequency of your microprocessor in MHz (for delays): 4 Change to:

From here, you could either type in the actual 2-Mhz ADAM speed by just typing "2" or you can "fudge" a little and type "4" which will work just fine. When you are done, you will again be asked:

Turbo Pascal general installation System
Choose installation item from the following:

Screen installation | Quit

Enter S or Q:

When you type "Q" to Quit, the new terminal escape codes will be written into the main program file for you. You've just finished your first program installation! Go ahead and try to boot the new program and see how it looks. If it's "not quite right", you now know how to go back in and try a different emulation.

PERFORMING AN "OVERLAY" INSTALLATION

An excellent example of an overlay type of installation would be represented by the *TERMINAL.Z80* file provided by Tony Morehen and Guy Cousineau with AJM Software's T-

DOS. This file is used to do a screen and hardware installation for various support utilities supplied with T-DOS. All that is required is that you have any good CP/M compatible text editor or word processor. You could use WordSTAR, VEDIT, ED.COM or my favorite, the latest release of VDE called ZDE. Just call up the file by typing the line:

VDE TERMINAL.Z80 /N (return). This line will tell your ADAM to load VDE, then to read in the file *TERMINAL.Z80* for editing in an (N)on-document mode. Besides the command line header of the VDE program, the text of the file will appear much like this example. I won't list the entire file here, but rather, just the parts you will need to be concerned with in patching for your terminal.

```
;
;sequence to turn inverse (or dim) video on
;
IVON$:    DB      ESC,'p',0,0,0,0    ;must be exactly 6 bytes - fill with 0
          DB      0                  ;end of string marker - do not change
;
;sequence to turn inverse (or dim) video off
;
IVOFF$:   DB      ESC,'q',0,0,0,0    ;must be exactly 6 bytes - fill with 0
          DB      0                  ;end of string marker - do not change
;
;sequence to clear screen and home cursor
;
CLEAR$:   DB      ESC,'E',0,0,0,0
          DB      0,0,0,0,0,0        ;must be exactly 12 bytes - fill with 0
          DB      0                  ;end of string marker - do not change
;
;sequence to clear from cursor to end of line
;
CEOL$:    DB      ESC,'K',0,0,0,0    ;must be exactly 6 bytes - fill with 0
          DB      0                  ;end of string marker - do not change
;
;sequence to clear from cursor to end of screen
;
CEOS$:    DB      ESC,'J',0,0,0,0    ;must be exactly 6 bytes - fill with 0
          DB      0                  ;end of string maker - do not change
;
;sequence to turn cursor on
;
CURON$:   DB      ESC,'y5',0,0,0     ;must be exactly 6 bytes - fill with 0
          DB      0                  ;end of string maker - do not change
;
;sequence to turn cursor off
;
CUROF$:   DB      ESC,'x5',0,0,0     ;must be exactly 6 bytes - fill with 0
          DB      0                  ;end of string maker - do not change
;
;sequence to insert line
;
INSLN$:   DB      ESC,'L',0,0,0,0    ;must be exactly 6 bytes - fill with 0
          DB      0                  ;end of string maker - do not change
;
;sequence to delete line
;
DELLN$:   DB      ESC,'M',0,0,0,0    ;must be exactly 6 bytes - fill with 0
```

```
        DB      0               ;end of string maker - do not change
;
;terminal init sequence
;
TINIT$: DB      0,0,0,0,0,0     ;must be exactly 6 bytes - fill with 0
        DB      0               ;end of string maker - do not change
;
;terminal deinit sequence
;
DINIT$: DB      0,0,0,0,0,0     ;must be exactly 6 bytes - fill with 0
        DB      0               ;end of string maker - do not change
;
;valid drives - set to 0 if present, 0FFH otherwise
;
VALDRV:
        DB      0               ;drive A
        DB      0               ;drive B
        DB      0               ;drive C
        DB      0               ;drive D
        DB      0FFH            ;drive E
        DB      0FFH            ;drive F
        DB      0FFH            ;drive G
        DB      0FFH            ;drive H
        DB      0FFH            ;drive I
        DB      0FFH            ;drive J
        DB      0FFH            ;drive K
        DB      0FFH            ;drive L
        DB      0FFH            ;drive M
        DB      0FFH            ;drive N
        DB      0FFH            ;drive O
        DB      0FFH            ;drive P
;
```

Now, I'll be the first to admit that at first glance, all of this data seems overwhelming. There is a great deal of data here that you will never need to worry about though. What you SHOULD look at are things such as the number of drives your system has hooked up to it. This can be disk drives, data drives and even a memory expander. The first drive is always drive A:, so looking at the last few pages of the text we see the area that assigns storage information. If you only have one data drive and a disk drive, a simple two drive system is all you need your program to support.

In the case of this two drive system, you would change the text I listed above from this:

```
;valid drives - set to 0 if present, 0FFH otherwise
;
VALDRV:
        DB      0               ;drive A
        DB      0               ;drive B
        DB      0               ;drive C
        DB      0               ;drive D
        DB      0FFH            ;drive E
        DB      0FFH            ;drive F
        DB      0FFH            ;drive G
        DB      0FFH            ;drive H
        DB      0FFH            ;drive I
```

```
        DB      0FFH            ;drive J
        DB      0FFH            ;drive K
        DB      0FFH            ;drive L
        DB      0FFH            ;drive M
        DB      0FFH            ;drive N
        DB      0FFH            ;drive O
        DB      0FFH            ;drive P
```

to this:

```
;valid drives - set to 0 if present, 0FFH otherwise
;
VALDRV:
        DB      0               ;drive A
        DB      0               ;drive B
        DB      0FFH            ;drive C
        DB      0FFH            ;drive D
        DB      0FFH            ;drive E
        DB      0FFH            ;drive F
        DB      0FFH            ;drive G
        DB      0FFH            ;drive H
        DB      0FFH            ;drive I
        DB      0FFH            ;drive J
        DB      0FFH            ;drive K
        DB      0FFH            ;drive L
        DB      0FFH            ;drive M
        DB      0FFH            ;drive N
        DB      0FFH            ;drive O
        DB      0FFH            ;drive P
```

As you can see, CP/M supports up to 16 different storage devices and your own system knows how to talk to any of them by its' own drive letter. It is your job in many cases, to tell your program about these drives that that IT can use the ones you have present on your system.

The case of the drive assignments is relatively easy to pick out among all the rest of that *TERMINAL.Z80* file, but what about the other important patching areas? Sometimes, you have to be a bit of a detective to deduce just what calling conventions are being used by any particular author. It has been my experience that as most programmers write the code for such things as inverse video, line inserts, screen clears, etc., they tend to label them in an easy to pick out fashion. The key is in our being able to look at that code and recognize these labels for what they are.

Suppose I had an overlay file such as the one above, but that has been patched for a Kaypro 1 computer system. To convert the code to match what my ADAM uses would require me to have a listing of all codes used by the Kaypro for video sequences. I won't list the whole file again as I did above, but just a few of the more important screen codes to give you an idea what we would be up against in converting that code to the Heath screen codes.

```
;sequence to turn inverse (or dim) video on
;
IVON$:  DB      ESC,'B0',0,0,0,0;must be exactly 6 bytes - fill with 0
        DB      0                       ;end of string marker - do not change
;
;sequence to turn inverse (or dim) video off
```

```
;
IVOFF$:   DB      ESC,'C0',0,0,0,0;must be exactly 6 bytes - fill with 0
          DB      0                ;end of string marker - do not change
;
;sequence to clear screen and home cursor
;
CLEAR$:   DB      1AH,0,0,0,0,0
          DB      0,0,0,0,0,0      ;must be exactly 12 bytes - fill with 0
          DB      0                ;end of string marker - do not change
```

Take a look at the sequence to clear the display screen and to home the cursor into the upper left corner of your screen. The Kaypro uses a CONTROL-Z command to implement this function. If you take a look at your CP/M manual's control codes listing, you can pick out the HEX code needed to represent the CONTROL-Z. In this case, that would be a 1A in hex. You could also use simply a decimal code, in this case 26. Since I prefer using HEX codes, I have to also tell my program that the 1A is a HEX value, this the 1AH you see in the label for *CLEAR$*.

The codes *IVON$* and *IVOFF$* represent inverse on and and inverse off respectively. Some programmers will refer to this sequence as reverse video represented by something like: *RVON$* or *RVOFF$*, while others may refer to *DIMON$* or *DIMOFF$* for DIM video on and off. In the case of the Kaypro, REVERSE video and DIM video or both supported as separate entities. You can even mix and match them into a DIM/INVERSE video by use of both codes!

The most important thing to remember here is that if you have a program designed for a particular computer, and you want to move this file over to use it on your ADAM, you MUST know the screen codes for the original computer as well as those for your ADAM.

When all of the codes have been changed to match the proper codes you want, you will want to save this new version by pressing your ESCAPE key followed by the X key. This will tell VDE or ZDE to save your new code, create a backup of the original and kick out of the word processing program and back to standard CP/M. Now that you are back to your "A0>" prompt, you will need to assemble the code into a HEX file. For this purpose, you can use the ZASM program, SLR, ASM or any of the many assembly programs available both commercially and in the Public Domain. If you want to use the ASM.COM program that came with your CP/M 2.2 package, you can do that after first renaming the file. ASM.COM doesn't know much about Z80 files as it is made to look for files ending in ASM.

In this case, you would want to type:

REN TERMINAL.ASM=TERMINAL.Z80 and press the **RETURN**. Under T-DOS, all that is needed is to type **REN TERMINAL.Z80 TERMINAL.ASM**.

As you can see, T-DOS has a much easier to follow format than CP/M. A CP/M system has to be told what you want the new name to be, an = sign, and then the old name of the file. T-DOS just lets you tell it what the old name is and what you want it's new name to be. It thinks about the name change in a fashion similar to the way your brain thinks about changing the name!

In the case of a file being assembled with ASM.COM (since that is the program supplied with CP/M 2.2 and Assembler!), we'll show it as that file type:ie/ *TERMINAL.ASM*

To do the actual assembly, simply type **ASM TERMINAL** and hit the return. There are

other "switches" you can use to eliminate the PRN file, send output to other drives, etc., which can be found in the appropriate section of the manual. For my purposes here, I'll just show you the easiest way to do this. ASM.COM will load into system memory. It will then read in all the code listed in the file *TERMINAL.ASM*. It already knows to look for a file that has that ASM extension. When it is done, you will find a couple of new files on your disk or data pack. There will be a *TERMINAL PRN* and also a *TERMINAL.HEX*. You can erase the PRN file if you wish to save space. It is this HEX file that we've been looking for all along!

Now, the final patch is done using a public domain program called MLOAD.COM. MLOAD is a modified version of the LOAD.COM program a CP/M package always seems to have. The program has a variety of uses that make it useful to keep around. It can take a HEX file and compile it into a COM file. It can also take a couple of files (or even several!) and combine them in the proper order to make an enhanced program out of a normal COM file. This last is what we are about to do.

Suppose we are trying to patch a program called CALENDAR.COM that just happens to recognize the calls named in our TERMINAL.Z80. A bit of editing has modified that Z80 file into something that can make the display look correct on our TV screen. After assembly, we can now merge the two files CALENDAR.COM and TERMINAL.HEX. To do this using MLOAD.COM, just type:

MLOAD CALENDAR.COM=CALENDAR.COM,TERMINAL.HEX and press the **RETURN** key. MLOAD will go out, load both files, merge the new values in the HEX file into the correct locations in CALENDAR.COM, even to the extent of replacing the older codes with the newer ones! The best way to keep things separate, though, would be to name a new version of the program to something different. That way the old one will stay as it is if things don't quite work out correctly.

If I wanted to merge the files, but save the result to a different file name, I'd call it something like CAL2.COM. In the case just mentioned, the program syntax would be:

MLOAD CAL2.COM=CALENDAR.COM,TERMINAL.HEX <RETURN>

PATCH EDITOR PROGRAM MODIFICATION

The final method I use to patch programs is, at least to me, the easiest of them all. To do this, you will need the excellent PD patch editor program called SPZ. This program has been revised a number of times, but any version from 3.3 to 3.7 are available to ADAM owners. This program will allow you to select a file from the disk directory to edit. Once selected, you will see a that a "page by page" program listing appears onscreen. The left side of the screen shows the address for each BYTE of the program and the right side shows the ASCII equivalent of those bytes.

At first boot up, SPZ will display it's version number, a list of it's commands, and then a list of the files on the current drive and user area. If there are more files present, press your HOME and DOWN ARROW keys at the same time to view the next page/s.

- SUPERZAP Version 3.7 -

^S	Cursor LEFT	E	Edit File	M	Map Directory
^D	Cursor RIGHT	T	Type File	L	Log New Disk
^E	Cursor UP	S	Select Track/Sector	U	Change User Number
^X	Cursor DOWN	^C	Next Directory Page	D	Drive Statistics
N	File Name Select	^R	Previous Directory Page	X	EXIT from Superzap

```
                    Directory List  -  A00:????????.???

180FIG   .COM       20MEGDIS.DIR       8080     .MAC       80BHD45 .COM
80NOHD45.COM        80TDOS45.COM       ADAMVLU  .COM       ADDRESS .
APPTS    .          ARK      .COM      ASCII    .          ASM     .COM
BART     .GRF       BASE     .COM      CD       .COM       CHRIS   .MSG
CHRISNEW.ADR        CLONE    .COM      CRLZ     .COM       CRUNCH  .COM
CU15     .COM       DALE     .NUM      DATE12   .COM       DATES   .
DB       .COM       DIR2     .COM      DIR4     .COM       DISK    .BIG
DISK     .GRF       DRIVES12.COM       DSKSZ24  .COM       EOS     .COM
```

From here, you can use your cursor keys to move around to the one file you wish to edit first. Once selected, simply refer to the options at the top of your screen to see what to do next. In our case, we will press E to begin an (E)dit of the program. I want to edit a version of SPZ (also known to many as SUPERZAP) that is currently configured for use with a Televideo 912 terminal. I'll have to use that HOME/D-Arrow combination a couple of times to find it on my directory, but once I see it listed, I just move to it with the normal arrow keys. With the pointer in front of that file's name (SPZ-TS1.COM), I press the E key and the display will change to this:

```
                          -   SUPERZAP Version 3.7   -

^X  Next Sector              S  Select Sector           A  Find ASCII String
^E  Previous Sector          E  Edit Sector             H  Find HEX Sequence
^R  Top of File              P  ScratchPad Mode         F  Exit to File List
^C  End of File                                         X  EXIT from SuperZap
 ScratchPad - Empty

 SELECT FUNCTION:

File Name       Access                Current Sector  Load Address
A00:SPZ-TS1 .COM  R/W                      0000          000100

000100   C3 6E 02 20  53 55 50 45  52 5A 41 50  20 33 20 20 |Cn. SUPERZAP 3
000110   20 20 66 6F  72 20 43 50  2F 4D 20 32  2E 32 20 20 |  for CP/M 2.2
000120   20 20 61 6E  64 20 43 50  2F 4D 20 33  2E 31 20 20 |  and CP/M 3.1
000130   20 20 57 2E  4D 2E 44 61  76 69 64 73  6F 6E 20 20 |  W.M.Davidson
000140   20 20 48 2E  4A 2E 53 68  65 6C 64 72  61 6B 65 20 |  H.J.Sheldrake
000150   02 1B 2B 00  00 00 00 00  02 1B 54 00  00 00 00 00 |..+.......T....
000160   02 1B 28 00  00 00 00 00  02 1B 29 00  00 00 00 00 |..(.......).....
000170   02 1B 3D 00  00 00 00 00  00 00 00 00  00 00 00 00 |..=.............
```

Okay, we've got a good starting base-line for our modification of the program over to one that will look proper on my Orphanware 80 column video unit (80-CVU). The unit emulates a Heath/Zenith H19 dumb terminal which has screen codes corresponding to those in my master chart listed earlier. What we need to do now is to find a way of learning the code values for the Televideo terminals. The codes will then need to be compared to the corresponding Heath 19 codes and plugged into our program.

There are files to be found on CompuServe, GE*nie, the A.W.A.U.G. BBS, The Cleveland Adam Exchange BBS and other's that also list a good cross-section of terminals. The file I like the best is one called simple TERMINAL.ARK. If you don't have access to such an all-too-useful text file, there is yet another way. If you have a copy of BORLAND's "TURBO PASCAL" program, you can compare codes between known and unknown terminals by using your word processor to look at the INSTALL.DAT file.

A free version can be found in a library called PDTINS.LBR in the CPM forum of CompuServe. This library contains an all-around COM file that's sole use is the installation of compiled PASCAL files into executable programs. Many top notch PASCAL programmers have created software they later released into the PUBLIC DOMAIN. The PDTINS.COM file let's us install that compiled program for our own terminal. It does this by reading from a menu (remember the menu I listed for installing SuperCALC earlier?) and when you are done selecting your terminal, it then reads the list of codes for that terminal from a file called PDTINS.DAT. This is almost the same file as the INSTALL.DAT mentioned above.

Since you have the Heath codes listed in the main chart, all you have to do is to look up the codes for HEATH/ZENITH in the PDTINS file and note their locations on the screen. Next, look on down to the codes for the Televideo (there may be several!) terminal. A quick look at those two pages of data will show you what codes MUST do what. In our case, we can see that the Televideo is using an ESC + to clear the screen and home the cursor. We can fix this easily by using SPZ to change the "+" to an "E".

Hey! Wait a minute! The Heath terminal will also respond to ^Z as a clear screen code! What if we wanted to substitute this for the ESC E combination? Well, it's very easy to do but I have an important code to point out. Look at he SPZ screen above. Look at the code section on the right (the ASCII side) and look at the "+" character. Now, look at it's HEX representation on the left side of the screen. You will see line 000150 as 02 1B 2B 00. In this line, the 02 tells the program how many bytes to look for to do any screen work. The 1B is the HEX value for our ADAM's ESCAPE key. Last but not least, the 2B is the HEX value for the "+".

The replacement of the ESC + (1B 2B) with a ^Z (1A) reduces the needed code byte number by one, so we will have to make a change in that 02 as well. The 2B will no longer be used, so it will be "nulled" out by changing it to a 00. The new line is: 01 1A 00.

How do I know what codes effect the screen display? It's partly a "trial by error" method, but my experience shows those screen codes to invariably begin with an ESC code of 1B hex. SPZ will let you scan for any 1B's. You merely look at each one of them in turn and see if any of them look like something found in one of your of terminal control code charts. SPZ happens to contain it's screen codes at the beginning of the program file. Is is for our purposes a good choice for a first time program to patch.

Okay now, with that taken care of, let's get back to our "alien" screen codes. A glance back at that SPZ screen shows four more characters that need to be resolved. The ones I'm looking at are the ones so strongly resembling our clear screen code. The ones I mean are the "T", "(", ")" and the "=" characters.

By looking at the PDTINS.DTA file, I've learned that the sequence of Escape T erases a line, escape (will start up "dim video", an escape) will stop the dim video and go back to "normal video".

The last of the characters is the "=" which we've learned is what the Televideo terminals use to set up direct cursor addressing. A use for this is that of keeping your lines and columns lined up. A lack of being able to read the current cursor position, saving it and later recalling it are just a few of the things handled by way of direct cursor addressing. In the case of our Heath/Zenith display, it can accept either an "ESC Y" or an "ESC =" for such a purpose, so we won't have to do anything to this particular code.

To clear a line from the cursor position, the HEATH requires the use of an "ESC K". To patch this one, we just do what we did for the ESC + code. Use SPZ's edit feature, move over to the right side of the screen, place the cursor over the T and type in a K to take it's place. That makes three of those pesky codes taken care of now!

Now we have a bit of a problem. The ESCAPE code followed by the (or) turn dim video on and off. The trouble here is that HEATH terminals can't do dim video! The Heath can, however, do INVERSE VIDEO and I find it much more pleasing to the eye. To achieve an inverse video effect, the Heath uses, as shown by our ESCAPE CODE chart, an "ESCAPE p" to turn on the inverse video and "ESCAPE q" to turn it back off. Using what we've learned so far, we simply replace the "(" character with a "p" and the ")" with a "q".

The last thing we need to do is to save all of these changes to a disk or data-pack. Doing this is the easiest task of all. Press the CONTROL key and then the "z" key and those changes will all be written to the original file. Press the "x" and SPZ will quit running and drop you back to the system prompt. Now try out your new version of the program to see if it all looks correct on your video display.

What I've tried to show you in the last few pages is that getting a program to run on the ADAM that has been written for some other computer isn't so difficult after all. You don't have to call in someone else to help you do it. YOU can do it all by yourself! Just stick to working on COPIES of the original file and you will never be disappointed. The first few tries at it may not work perfect, but you can still try again on another copy. Sometimes you may have to go on and look for other codes to patch. The SPZ example certainly has a few more. I'll leave those to YOU!

Ronald W. Collins

****

FROM THE ADAM XCHANGE BBS COMES THE FOLLOWING:

UNDELETE.COM
A SMART FILE UNDELETE UTILITY
By Guy Cousineau

1990-01-28 1.2	Fixed for SECTOR PER TRACK > 255
	Fixed wild card mask to find high-bit-set
	Fixed double show of files > 512K
	Moved CP/M 2.x check to program start
	Incorporated Disk reset routine
	Changed U=UNDELETE to T=TAG
	Speeded up duplicates check
	Added RAMTOP to protect from HUGE directories
1990-01-11 1.1	Fixed sector translate bug
	Fixed for disks with more than 16 SPT
	Added wild card mask to screen files
1989-08-02 1.0	First released to public domain

Special thanks to Bridger Mitchell for a selective file reset utility for logging in fixed drives. His full subroutine is included intact and is fully transportable to any other program. See the large chunk of code written entirely in lowercase.

Thanks also to Howard Goldstein who was of invaluable help in debugging the preliminary versions of this utility.

PREAMBLE

First a few explanations about deleted files and directory structure. CP/M maintains a 32-byte directory entries for files. The first 16 bytes are similar to a File Control Block (FCB):

```
FCB OFFSET          0 1 2 3 4 5 6 7 8 9 A B C D E F
CONTENTS  d n n n n n n n n t t t e s1s2rc
```

The drive code goes in position 0
bytes 1-8 contain the file name
bytes 9-11 contain the file type
byte 12 contains the extent
bytes 13-14 are internal control bytes s1 and s2
byte 15 contains the number of records in this extent

So what's an extent? Consider it as the number of chapters in the files, and the record count as the number of pages in this chapter. The working of extents is a fairly complex exercise depending on your disk architecture. The difference between an FCB and a directory entry is the first byte. Once the directory is written to the disk, there is no need for the drive code so instead, the BDOS places the user number in byte 0.

When a file is deleted, the BDOS replaces that user number with an E5. This is the signal for a deleted file entry which can be re-used by another file. Unfortunately, the E5 also wipes out the old user number.

UNERA.COM was a welcome addition to the CP/M public domain and was very useful in recovering accidentally deleted files. It looked for files with the E5 code in byte 0 and replaced them with the default user number. It had, however, a few drawbacks in that it allowed unerasing files which would duplicate entries already in the current DU and unerased files which overlapped the same allocation blocks as other active files.

Hence the creation of UNDELETE.COM. This utility works in the following fashion:

1 find all deleted files (or mask provided on command line)
2 mask out files with no EXTENT 0 (incomplete)
3 mask out files with allocation overlaps
4 mask out files with the same name as active files (slow part)
5 prompt for tagging of files to recover
6 undelete the tagged files

SYNTAX: UNDELETE [d:][afn]

DETAILED EXPLANATIONS

1
If no drive is specified the current drive is used. The program LOGS the selected drive out and then back in to force a directory read. Each directory record is read and scanned for deleted files, building table entries in RAM for each found file. If a file name was specified on the command line, table entries are created only for deleted files matching the AFN supplied. If too many files are found, the program will abort. Thsi might only happen on a 64K directory with 1/2 its files deleted.

2
Print first report skipping all entries corresponding to further extents of a file. This fashion of reporting effectively masks out large files for which extent 0 has been overwritten but other extents still exist in the directory.

3
The table is re-scanned and each allocation block checked against a copy of the allocation vector. If there is overlap, the file flag is set to disable recovery. If the allocation is unused, it is set to used to prevent recovery of other files which may use the same allocation. A second report is then printed.

4
Check the current drive/user for matches of the deleted files remaining in the table. These files may be undeleted later if the corresponding active file is renamed. A third report is then printed showing only the files which are safe to undelete.

5
Print the files out one by one and ask the user to tag them with a 't' or a 'T'. After the files are tagged, the user is asked for confirmation: a 'u' proceeds and a CONTROL-C aborts. In order to abort the tagging process and jump to the CONFIRM stage, just press ^C.

At every stage of the analysis, a printout is given of the FILES FOUND SO FAR. This way, you can see those files you want to recover but might be unable to because of internal conflicts. If at any time, the number of recoverable files becomes 0, the program aborts. A ^C will also abort the program at the CONTINUE prompts.

USE CONSIDERATION

Although this utility is smarter than UNERA.COM it can fail if a file with more than one directory entry is missing its second entry. Under normal directory operations, this is an impossible situation since directory entries are filled in from the start of the directory. Thus if extent 1 is missing then extent 0 must be missing as well.

The ALLOCATION overlap can also fail if a sort-and-pack program was used to re-order the directory entries. Under normal operations, it is impossible for the second of 2 deleted files (using the same allocation blocks) to be the UNDELETEABLE one. With SAP, there is no way of accurately determining which file can be undeleted.

Both the above checks can be thrown out of whack if a file was manually deleted by a disk editor (like DUU or ZAP). A variety of other unusual happenings may affect program behaviour.....tread carefully.

If there are 2 copies of the same deleted file, they will show up as 2 entries in the TAG menu. To recover both and check which is the most current, start by tagging only one version and rename it to another UNUSED name. Run UNDELETE again and tag the second version. Check them both out and delete the one you don't want to keep.

How can this happen you say? Pull out your text editor (the one that creates .BAK files) and follow these steps: (My editor is VDE)

```
VDE a.a
this is a.a version 1                [type this in]
                                     [exit]
VDE a.a                              [re-edit the file]
```

this is a.a version 2 [change the 1 to 2]
[exit]

Now I have just what I expect; 2 versions of the same file, but with different names. But I want to go back to version 1; so as I often do:

```
ERA a.a
REN a.bak a.a
```

After typing or doing whatever with a.a, I delete it as well. Later, I want to get it back and pull out UNDELETE which reports 2 files called a.a and both recoverable. If I tag them both, I can be in trouble. Although both are undeleted, the BDOS will only be able to find the FIRST one when I try to open it. If I try a rename or delete, it will find both and confuse me even more. So I should tag one, rename it, and run undelete again.

Although UNDELETE is smarter than UNERA, you should verify all recovered files before using. Particularly in the case of .COM files, it is wise to tread safely with back-up disks when testing the recovered file.

Since UNDELETE recovers files to the current user area, you should select the user number which is appropriate to the recovery attempt. Hey, you can even use UNDELETE to move a file from one user to the next by deleteing it from one area, logging into another user, and undeleteing it.

The utility has been tested on a CP/M 2.2 system and should behave correctly on any Z-80 CP/M 2.x. It has been tested with 1,2, and 4 K allocation blocks on media with 1 and 2 byte allocations. Although it has performed with no problems on these media, it is not guaranteed to work in non-standard environments. The utility will only run on a BDOS which reports a "2?" HEX when asked for version. CP/M 3.x does not like programmers writing directly to disk via BIOS calls.
Use the program in good health, please send bug reports to:

Guy Cousineau
1059 Hindley Street
OTTAWA Canada
K2B 5L9 OTTAWA RCPM (613) 952-2289

****

The Fine Print

An Update on the ADAM SURVIVAL GUIDE

An Adam General Information Desk Book containing all you need to Survive with our Orphaned Adam a how to find other information you might need.

The preliminary details have undergone several revisions and continue to be revised in our attempt to publish what can prove to really be an ADAM SURVIVAL GUIDE.

The ADAM SURVIVAL GUIDE will have 34 separate chapters on such subjects as Important addresses, Service/Repairs, User Groups, Publications,newsletters, Dealers, Software, Hardware, Graphics, CPM, TDOS, Smart Basic I & II, LOGO, Games, Speedywrite, MIDI, and others. We are looking at approx. 200 pages of Adam information.

Chapter Editors include: Tom Keene, Dean Roades, Rich Clee, Ron Mitchell, Phil Kosowsky, Eric Danz, Ron Collins, Bart Lynch, Guy-R. La Forest, Guy Cousineau, **Peter Hartzler, Bob Blair,** Mel Ostler, Rich Lefko, and several others.

TO BE AVAILABLE NOT LATER THAN AUGUST 1, 1991 AT ADAMCON 03.

PRICE, still to be determined as we are still obtaining printing costs. Printing costs for 200 pages are large.

The 1986 Adam Resource Directory had 84 pages and cost $ 13.95. The Adam Survival Guide will be approx. 200 pages or 2 1/2 times as large and we are attempting to keep the cost at $20.00 PLUS shipping.

Profits, if any, will be used for future A.N.N. and Adam projects. (It will go back into furthering Adam).

SUPPORT ADAM, SUPPORT A.N.N. ----- PURCHASE THE ADAM SURVIVAL GUIDE.

VERY IMPORTANT WE NEED DEALER ADVERTISEMENTS FOR INITIAL PRINTING COSTS AND RESPONSE HAS NOT BEEN AS GOOD AS WE WOULD LIKE. MEL OSTLER (ROAD RUNNER PUBLICATIONS), NORM CASTRO, ELLIAM ASSOCIATES HAVE PAID FOR THEIR ADS AND WE HAVE COMMITTMENTS FROM PAT HERRINGTON (EZOD GRAPHICS), STEVE MAJOR (ADAM CONNECTION), AND ERIC DANZ (ADAMZAP). LET THESE DEALERS KNOW YOU APPRECIATE THEIR SUPPORT OF ADAM, ANN & ASG. PATRONIZE THEM AND TELL YOUR FRIENDS TO DO SO.

When you write to or buy from an Adam Dealer, ASK THEM TO ADVERTISE IN THE ADAM SURVIVAL GUIDE AND TO SUPPORT IT.

SEVERAL OTHER PUBLICATION EFFORTS ARE BEING CONSIDERED BUT WILL DEPEND ON THE SUCCESS OF THE ADAM SURVIVAL GUIDE.

As usual A.N.N. projects are still being handled by a small group of volunteers at their own expense, costs and time. We always need and welcome volunteers to work on A.N.N. projects, from things as simple as typing hardcopy to disk or tape, sending out form letters to more technical and detailed endeavors.

If interested in working on A.N.N. projects please contact me, giving me some information on your background, Adam setup, Adam interests, Adam/computer skills, if any.

LETS KEEP ADAM AND A.N.N. ALIVE: SUPPORT THE ADAM SURVIVAL GUIDE.

Very best wishes, Barry A. Wilson, 904 No. Warson Rd., St.Louis MO 63132

FOR SALE

Roller Controller with Slither Cartridge -- Like new	$30.00
Super Sketch with software cartridge and manual for Colecovision or Adam "A simple way to create super video graphics"	30.00
Adam computer with keyboard, manuals, data drive and printer	125.00
Coleco modem with software, card and manual	30.00
Expansion memory, 64K	30.00
Printer interface for dot matrix printer with cable	30.00
Orphanware 80 column monitor interface -- Adam to composite 80 col monitor (with proper cable, can be used with FALCO or other serial terminal)	50.00
Supersketch for Commodore with manual -- no software	10.00

Add packing and shipping charges to above items

Tom Barrett
6819 Rosemont Drive
McLean, VA 22101
703-356-6180

****

AWAUG
c/o Tom Barrett
6819 Rosemont Drive
McLean, VA 22101

*** FIRST CLASS MAIL ***