NYBBLES & BYTES

Jan & Feb
1991

N.W.P.A.C. ATARI 8-BIT CLUB NEWSLETTER

NIBBLES & BYTES

This Newsletter is written and published Bi-Monthly by the North West Phoenix Atari Connection (NWPAC). A non-profit organization devoted to the exchange of information concerning all Atari 8-Bit computers. NWPAC is not affiliated with Atari Corp..

Procedures for Submitting Newsletter Articles for Nibbles & Bytes

Articles are welcomed and can be up-loaded to PAUGS. Or they can be given to the newsletter editor, or any club officer at the regular meetings. If you are unable to attend the meetings and don't have a modem, you can send them to:

Newsletter Editor
P.O. Box 36364
Phoenix, Az. 85067

All text should be Up-Loaded in true ASCII format. AtariWriter+ or TextPro is suggested for XL/XE's.

Pictures, illustrations, etc. can also be submitted. Print Shop, MicroPainter, Koala or any other 62 sector pictures are preferred.

Deadline for all materials to be included in the Newsletter, must be in 20 days before date of issue.

Nibbles & Bytes Editor

Mike Barnes [Editor-n-Chief]
(894-9156)

This issue was produced on an Atari 130XE with News Station, and Printed on a Star NX1000.

Advertising Rates

Classified Ad's of a non-commercial nature are $1.00 per ad, to all current members.

Camera-ready commercial rates are:

Full Page:	$30.00
Half Page:	$20.00
Quarter Page:	$10.00
Business Card:	$5.00

NWPAC OFFICERS

President	Charles Lake 269-1209 (BBS)
Vice President	Carl Thornton 833-0357
Recording Secretary	David Farley 821-1197
Membership Secretary	Dan Varga 963-3814
Treasurer	John Collins
Disk Librarian	Merle Barnes 894-9156
Magazine Librarian	Ron Gibson

Area Code = (602)

Technical Assistance

The following NWPAC members have Offered their assitance to fellow members who may be having difficulties with thier Atari computers, or just have questions about software, hardware, or programming. If you are having problems, feel free to call anyone on this list for help.

Programming		
Charles Lake	BBS #	269-1209
Paul O. Parks	Voice	278-2375
General Info		
Charles Lake	BBS #	269-1209
Paul O. Parks	Voice	278-2375
Merle Barnes	Voice	894-9156
Repair		
Jerry Harris	Voice	846-5812
Hardware		
Charles Lake	BBS #	269-1209
Carl Thornton	Voice	833-0357
Software		
Paul O. Parks	Voice	278-2375
Charles Lake	BBS #	269-1209

Assistance

If you would like to offer your assistance also, contact the Editor at 894-9156 and get your name on the list..

CLUB BBS

(PAUGS)
Phoenix Atari User Group System

NWPAC provides a BBS system for it's members. PAUGS (278-8505) is operated 24 hours a day, 7 days a week, and hosts 300/1200/2400 baud rates. BBS usage is open to the public with private message bases and filesections for club members only.

*SysOp: Les Demaio
*Co-SysOp: Charles Lake
*Co-SysOp: Anthony Woods

ULTRASPEED + Part II
OPERATING SYSTEM
BY JEFF KYLE

Reprinted from Atari Interface
January 1989

The OS also gives you flexible command over the high speed. Pressing SELECT-OPTION will disable the high speed, and START-OPTION will reenable it. Also, holding SHIFT or SELECT when formatting will format the disk in standard slow skew instead of high speed skew.

If you've expanded your XL or XE to 256K or beyond, the US+ OS will automatically configure your RAM as a standard, configurable single or double density disk. The nice thing about this is being able to have a standard RAMdisk, compatable with anything.

It's easy to format it and write out DOS to it, so whenever you want DOS, you can boot from RAM. Also, it is set up to use the standard 130XE 128K RAM banks last. What this means is that, if you have 320K or more, you can have a full RAMdisk and run a program that uses 128K (such as Video Blitz, 130XE Koala viewers, Atari Writer Plus, etc) without worrying about damaging anything in RAM.

As in the XLF+ OS, the US+ OS reverses the OPTION key, speed up the keyboard response, and darkens the background color. However, now you may change some of these along with doing some of the things the 1200XL owners could do with their function keys, with your US+ OS, by using CONTROL-n or SHIFT-CONTROL-n. This is what you can do:

CONTROL-8 will lock or unlock your keyboard, in case you don't want people messing with it.

CONTROL-9 will toggle the internal BASIC. This command only goes into effect upon reset.

CONTROL-0 will restore the normal background colors and slow cursor.

SHIFT-CONTROL-0 will turn them back

SHIFT-CONTROL-7 will toggle the disk I/O noise on and off.

SHIFT-CONTROL-8 will turn the screen DMA off, speeding up most processes by 30% or so. Any other key will enable it, preferable SHIFT-CONTROL-A, which causes no character.

SHIFT-CONTROL-9 toggles the keyboard click.

There are other custom functions with the US+ OS; one is a modification to press SHIFT-CONTROL-Clear instead of SHIFT or CONTROL-Clear to clear the screen. the only problem is that with Action!, SHIFT-CONTROL-Clear will normally bring you to the left side of the screen. This will no longer work in the OS. If you need it, just flip to the standard XL/XE OS, then switch back after you've used the key.

Also you may now press CONTROL-4,5,6,7 as well as CONTROL -, =, +, and * to move the cursor up, down, left, and right. Also by pressing CONTROL-Caps, you not only can use the normal graphics characters, but also use the cursor keys without holding down CONTROL.

By either pressing HELP-RESET or pressing SHIFT-ESC then pressing RESET, you can force a cold start of the computer. This is useful when you've got important files in the RAMdisk and the computer won't let you RESET out from a program normally. One powerful feature of the OS is it's ability to reconfigure drives for different numbers and boot from any drive, including RAMdisk. When you hold down START after a RESET (warm or cold start), you'll enter a menu that has lists of drives 1-9 and what each is assigned as.

Normally, they are all assigned to themselves, except drive 4 which is normally Ramdisk. Say you wanted drive 4 to be drive 1 and drive 1 to be drive 4. You'd press RESET and START. You then press C for Configure, enter the original drive (1), then the drive you want it to be (4). Then you repeat the procedure for drive 4. It's that easy! With this menu, you can also change your RAMdisk number just with R for RAMdisk, then enter whatever drive you want it to be.

Also built into the menu is a small sector copier, mainly for drive-to-RAMdisk or RAMdisk-to-drive copying. It will read the density, format the destination in the density, and copy the disk a sector at a time. If you want, you can do normal drive-to-drive copies, but because of the one-at-a-time approach, it won't be as fast as a standard sector copier.

If you've entered the menu from a coldstart, you may also temporarily boot from any drive, just by pressing the drive number. This will swap the drives. By pressing SHIFT-CONTROL-6, you can "unswap" the drives and restore the drives to the original configuration. This won't reset everything, it just swaps back the drive you booted from. For instance, if you booted from drive 3, you could still use drive 1 if you referred to it as drive 3. Pressing SHIFT-CONTROL-6

will reset it so that drive 3 is drive 3 again and drive 1 is drive 1 again.

Occasionally in the standard OS, an OSS "supercartridge" can be ignored due to it's bank selecting. The US+OS sends a command to "wake it up" to eliminate this problem. You can also easily install an external switch to swap the RAMdisk and drive 1. With this switch, you can make RAMdisk drive 1 at any time.

Unfortunately, if you use SpartaDOS, the reconfiguring will have no effect due to it's using custom routines, bypassing the OS. Also the Virtuoso player will always go to the "real" drive 1 instead of whatever is configured as drive 1. Also SpartaDOS will not boot form RAMdisk, allow the keyboard functions (without a KEY OFF command), or allow high speed with the XF551 drive. But if you MUST use this DOS, it is easily modifiable with a built-in patch called by a simple USR command that will modify SpartaDOS to allow these things.

The new OS has removed the standard Self-Test routines and put in a RAM check supporting up to two megabytes. This is also the FASTest RAM checker I've ever seen. Also, in the same vein, the US+ OS frees up pin 17 on the PIA for easier one-and two-meg upgrades. Doing so will disable the self test and the standard XL/XE OS, but it will work.

Before the OS, there was a problem if you turned your machine off and back on very quickly, because of the way the high-density RAM chips retain their memory for a few seconds after the loss of power. With the new OS, this problem has been eliminated.

The US+ OS also supports a separately sold RAMdisk write protect switch which will protect either your RAMdisk memory or all of the extended memory from writes.

The documentation that comes with the OS is satisfactory. It explains all the functions in detail with many examples. It explains the functions and the installation completely. It would be nice if it told more of the technical information, but it is better than most information that comes with other custom OSs.

Installation? I can't really say about the difficulty as I had mine installed by the author. But looking through the instructions, mainly it should be easy on the 800XL. On the XE's most of the time the OS is not socketed in, but soldered in. If it is soldered in, you must desolder the OS, which requires a bit of soldering skill. Other than that, it is mainly soldering a few wires to some of the chips, plus drilling a 1/4" hole for the 3-position switch to switch between the OS's.

And that's the OS! I have found very few compatability problems, but when they do arrise, it's as easy as switching back in the regular XL/XE OS. Having all the special functions available is is handy, as well as being able to boot from RAM.

However, there have been problems when switching from the standard XL/XE OS to the XL Fix+OS. Because of different locations, switching can occasionaly make a program point to something that can screw up your RAMdisk. This is rare, but is a good idea to back up your RAMdisk before just jumping into the XLF+OS.

Other than that, I've had very few problems with the OS. It works as stated, and the keyboard functions work with most programs. At $69.95, it's a great buy. So if you'd like high speed with your drive, better RAMdisk control, or just plain more power over your computer, don't wait. Buy this OS!

ED-PLEASE CHECK AVAILABILITY, AND CURRENT PRICING.

TREASURY REPORT

MEMBERSHIP REPORT
BY: DAN VARGA

Attendance:
24 Members
02 Visitors
01 New Member
02 Renewals
06 Dropped From List

Money made:
$15.00 New Member
$30.00 Renewals
$20.00 Raffle Tickets
$65.00 Total

Raffle winners: Herb Craft - Took home Silent Service
John Collins - Won the 1020 Plotter

NEW OWNER'S COLUMN
by David Plotkin
Reprinted from Antic Magazine
June 1986, Vol.5 #2

LESSON 4: NESTED LOOPS

Last lesson, we explained how to use IF/THEN statements for making decisions, and how to use FOR/NEXT statements to execute loop. This month we are going to talk some more about these two sets of commands, as well as some commands for making IF/THEN and FOR/NEXT more flexible.

The IF/THEN statement is used to test IF a condition is true, and THEN take the appropriate action. We are not limited to just testing whether two variables are equal. A whole range of tests can be performed. You may test for the following conditions:

=	Variables are equal to each other
<	First variable is less than the second variable or constant
<>	First variable is NOT equal to the second variable or constant
>=	First variable is greater than or equal to the second variable or constant
<=	First variable is less than or equal to the second variable or constant

With this many options available, the IF/THEN statement can be used to test nearly any set of conditions. For example:

```
10 IF (XXX)=4) THEN .....
   (greater than or equal to)
20 IF (XXX<>YYY) THEN....
   (not equal to)
30 IF (XXX<1) THEN.......
   (less than)
```

NESTING

Both the IF and the THEN must be contained within the SAME program line. But the IF/THEN statement can be NESTED. That is, an IF/THEN statement may contain another IF/THEN statement, up to the limit imposed by the length of the line:

```
10 IF XXX=5 THEN IF YYY=6
   THEN IF ZZZ=7 THEN PRINT
   "ALL CORRECT"
```

The words ALL CORRECT will be printed only if ALL THREE conditions are met. If any of the conditions are false--that is, if any of the variables are NOT equal to the number in the equation, then the rest of the statement will not be executed.

There is a more efficient way to test for multiple conditions than using multiple IF/THEN statements--the AND and OR commands which were mentioned last month. The AND and OR commands will test multiple conditions, two at a time, to determine whether the combination of the conditions is true or false. For the AND statement, the result is true only if BOTH statements are true:

AND "TRUTH TABLE"

statement	statement	result
True	True	True
True	False	False
False	True	False
False	False	False

An example of using the AND statement in a program might be:

```
10 IF (XXX=5 AND YYY=6) THEN
   PRINT "BOTH CORRECT"
```

The words BOTH CORRECT will print only if XXX=5 and YYY=6 (both are true). If either one is equal to something else (one is false), the the words will not be printed.

The OR statement words somewhat differently. The result is true if EITHER ONE is true:

statement	statement	result
True	True	True
True	False	True
False	True	True
False	False	False

An example of using OR might be:

```
10 IF (XXX=5 OR YYY=6)
   THEN PRINT "ONE OR BOTH
   ARE CORRECT"
```

The words ONE OR BOTH ARE CORRECT will print out UNLESS neither variable is equal to the appropriate value. If either one is equal (one is true), then the statement will print out.

Although the AND and OR commands test conditions for true or false two at a time, they can be used to test more than two conditions. Take

this example:

```
10 IF (XXX=5 AND YYY=6
   AND ZZZ=8 AND BBB=9)
   THEN PRINT "ALL CORRECT"
```

This line will be evaluated by first testing whether XXX=5 and YYY=6. If the result is true (they are both equal), then the true result will be tested against the next condition (ZZZ=8). Each intermediate result is used as one of the two conditions to test with the next condition.

Clearly, in this example, any of the statements being false will lead the whole equation to be false and the

words will not be printed.

AND and OR can be combined in the same statement. Again the conditions are evaluated two by two, with each intermediate result used to evaluate the next condition. Assuming that XXX=5, YYY=6, and ZZZ=7:

```
10 IF (XXX=5 OR YYY=7)
   AND (YYY=8 OR ZZZ=7)
   THEN PRINT "WHAT A TEST"
```

This statement evaluates a true, and the words are printed. XXX is equal to 5, so the first combination (XXX=5 or YYY=7) is true. ZZZ is equal to 7, so the second combination (YYY=8 OR ZZZ=7) is also true. ANDing the two true conditions together(TRUE AND TRUE) is also true. The use of AND and OR is a very powerful tool in making program decisions, so you should be comfortable with their use. If you aren't too sure that you can determine when a statementis true or false, check this month's type-in program for quiz of whether various complex program statements are true of false.

The final command that is useful in making decisions in a program is NOT. The NOT command takes the opposite of of an evaluation's result. That is, if a statement evalutes as true, then NOT will evaluate the statement as false:

```
10 IF NOT (XXX>1 AND YYY=3)
   THEN.....
```

If XXX=2 and YYY=3, so that the statement (XXX>1 and YYY=3) is true, then the above line wil not execute the part of the statement after "THEN" because NOT takes the true statement and makes it false.

FOR/NEXT NESTING

Last month I explained FOR/NEXT

loops. There will be times when you want to change two or more variables at the same time. ONe way to do this is to NEST two or more FOR/NEXT loops--place one loop inside another:

```
10 FOR LOOP1=100 TO 200
20 FOR LOOP2=15 TO 4 STEP-1
30 SOUND 0,LOOOP1,10,LOOP2
40 NEXT LOOP2
50 NEXT LOOP1
```

This example will play a type of whistle on your Atari. Let's see how this works. Line 10 initializes the first loop variable (LOOP1). Line 20 initializes the second loop variable (LOOP2). Line 30 plays the sound, using the values of LOOP1 and LOOP2 as parameters. Line 40 represents the termination of the LOOP2 loop. Finally, line 50 terminates the LOOP1 loop.

There are some important things to notice about this example. When nesting, each inner loop must reside entirely within an outer loop. In this case, the inner loop (LOOP2) starts and ends within the outer loop (LOOP1). Each time the outer loop executes once, the inner loop executes through the entire range of the variable specified in the FOR statement.

If an inner loop is not contained entirely within an outer loop, ann error will result. This can happen accidentally in a program when you doing a lot jumping around. (GOTOs will be discussed in a future lesson.) When you use nested loops, you have to be careful about what occurs in the inner loop. For example, if your inner loop MODIFIES the variable used in the outer loop, you may exit before you want to, or you may never exit at all:

```
10 FOR LOOP1=0 TO 10
20 FOR LOOP2=100 TO 200:
   LOOP1=0:NEXT LOOP2
30 NEXT LOOP1
```

This program will run forever (or until you press the [BREAK] key) because theinner loop is setting LOOP1 equal to zero each time it is executed. The outer loop will increment LOOP1 to 1, but it will never reach 10. This illustrates that special care must be taken when you are programming nested loops.

FOR/NEXT EXITS

Some of the programming commands to

be covered in future lessons will tell you how to jump from one program line to another during the run of the program. Obviously, such commands could be used to jump out of a FOR/NEXT loop before the variable has reached its limiting value:

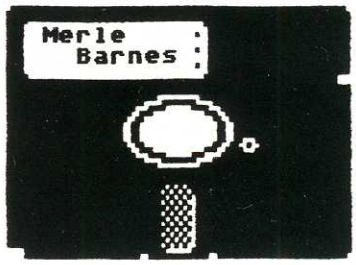
```
10 FOR LOOP=1 TO 10
20 GOTO 40:REM(sends the program
   to line 40, skipping line 30)
30 NEXT LOOP
40 REM Pick up here
```

This is BAD programming practice! If you leave a loop without completing it and executing the NEXT statement, your computer will not know that the loop is finished. The part of memory in your computer which keeps track of NEXT statements will gradually fill up with uncompleted NEXT statements, eventually leading to an OUT OF MEMORY error. This can happen even though you have plenty of regular memory left. There is a much better way to exit a loop without going through it as many times as the FOR statement specifies. Have the statements in the loop modify the loop variable so that it is outside the range specified in the FOR statement:

```
10 FOR LOOP=1 TO 10
20 PRINT LOOP:IF LOOP>5
   THEN LOOP=11
30 NEXT LOOP
```

This set of statements will execute until LOOP=6. Then the IF/THEN statement in line 20 takes over and makes LOOP=11, which is outside the range specified in line 10. Since LOOP is outside the range, the program will exit the loop and "fall through" to the program line folowing the NEXT LOOP statement. This method of proper exiting a loop cleans up your computer memory and doesn't lead to the problem discussed earlier.

1991



Disk Librarian

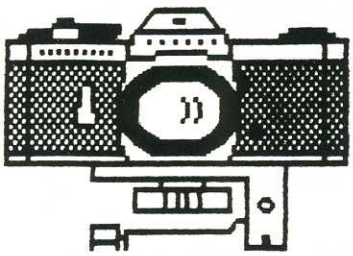
Here it is the end of another year. Seems like only yesterday and I was writing the first article for 1990. Now how things change I was the President and looking for a big year for NWPAC. But things do get changed around and when Paul wanted to step down as the Librarian and no one would volunteer for the job than a decision had to be made and I felt that the Disk Librarian was more important to the club than President and with that I took over Librarian; but, Paul was and still is the best we have had. I would like to thank some of he people who have donated disks into the Library so new programs are available to all of you. Working our way upward - Dale Wooster donated a disk; David Farley, John Collins and Charles Lake donated two disk; Carl Thornton gave five disks; Bob Woodson added 12 more and than I put in 14. The term disk meaning one side of a disk. This added a few disk to the Library by years end.

What does 1991 have in store for the Library. Well with a little luck, I will get over with Paul, at least by the end of 91, and learn all of the good things that he knows to try and keep the Library in the same condition that it was when he was running it. Of course, much of this is based on the future which is uncertain to say the least and it could be that Paul will be called upon to break in a new librarian. Whatever happens I have really enjoyed this past year as both President and Librarian and it has been a pleasure meeting people with the same interest that I have and being able to contribute a little back to the club that I have gotten so much out of the past 3 years.

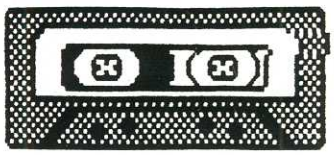
Lastly, I would like to wish everyone a Happy New Year and that it be full of happiness and computing.

A special wish for a speedy recovery by Bob Woodson and hope he is out of the hospital to read this newsletter.

Happy New Year



CLUB MINUTES



DECEMBER 1990

BY CARL THORNTON

Minutes for the December NWPAC meeting.

9:00 - 9:30am Setup and Chat session.
9:30 Merle Barnes (Disk Librarian) was asked to temporarely take over the meeting, so President Charles Lake could wake up the Recording Secretary.
Discussion about moving away from the church, and finding a new meeting place.

9:50 - 10:00am President Charles Lake arrived after talking to Carl Thornton on the Phone.. Charles asked for help, in trying to find another place for our meetings, a few suggestions were submitted.

Charles gave the Treasury report, and the BBS report. Balance as of November was $625.07, $181.00 was deposited, for a ending balance of $806.07.. On the BBS, there are now 4 networked message bases, with Battlestar..

Dan Varga gave the Membership report.

Merle Barnes gave the Disk Librarian report.

Mike Barnes gave the Newsletter Report.

Nancy Gibson gave the Publication Report. She's trying to organise the entire magazine library. She made everyone aware that if they needed a certain issue, to give her a call, and she'd make sure it was at the next meeting.

Charles mentioned the purchase of NewsStation to take place within the next month, funds have already been allocated for the purchase.

Networking with Iron Horse, Charles mentioned the possibility of networking with Iron Horse BBS. He reviewed the plans, and asked for club approval to do so.. A few questions were asked by Dave Farley, about Iron Horses policy concerning DL/ULing of files.. Charles described the policy Iron Horse has, and explained why David had the problem. Merle Barnes proposed a 2 Month trial period of the Networking, that was aggreed to by the members.

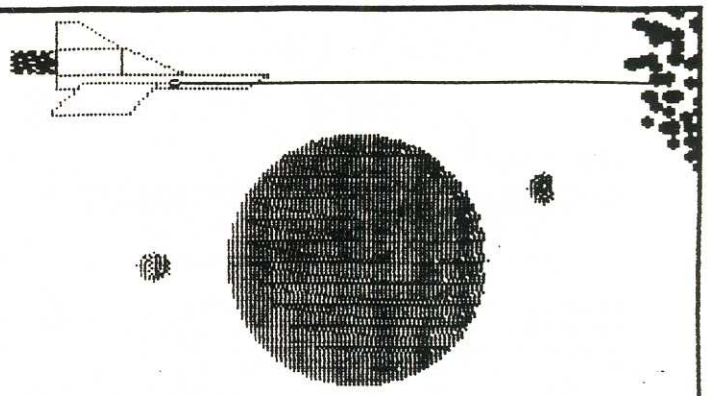
Open discussion period will start in Janurary, to help with problems in a roundtable type form.

Charles then continued to stall, then decided to carry out the Election.. Last minute Nominees were requested, none found. President Charles Lake then continued the Elections, Charles Lake was voted in for a second term. The Nominee for Vice President, Carl Thornton, hadn't arrived, but was voted into office. As a little joke, it was aggreed that everyone would mention that Bart Simpson was elected into that position, because of the Supplement statement by Carl. A motion was made to vote all of the other officers into office, because they were unopposed. The following were the unopposed candidates, Dan Varga was voted into the Membership Secretary's Job, David Farley was voted it as Recording Secretary, and John Collins was elected as the Treasurer..

Since the Recording Secretary wasn't present, Charles decided to have Mike Barnes do his demo first. Mikes demo was of StarFleet 1. Afterwards, Carl did a Demo of SpartaDOSX.
meeting adjourned!!!!

Colonial Warrior's Corner

By Charles Lake
President of
N.W.P.A.C.



Well to start this off, I like to thank all of you for your support this year of 1990, as NWPAC is going into its 7th year in helping its members by providing information and help with the atari 8 bit computer systems. Also our disk library has really grown thanks to Merle Barnes, as every month he & someother members put more PD files into the library. Thats what makes our club's disk library basicly the best in the nation of atari clubs.

NWPAC first started back in 1984, and its great to see a atari 8 bit club keep on truckin, being that the company that made the atari abandon us and others keep saying the 8 bit is dead. I don't think so, because our club is staying strong and will keep on as long as we stay together and support each other, and just the fact that our club is still alive proves that the atari 8 bit is not dead, but still alive.

The software from big companies are not being put out as much anymore but people that have 8 bits atari computers are programing their own and putting them out as shareware. Shareware means that you try out the program and if its useful or you like it, you donate money to the author to support more programs to come out or a updated version of that program. Remember support shareware, its basicly all we have now. this year several programs came out as shareware, that is bobterm 1.2, several versions of textpros, and others. I myself will be putting out one called Colonial Colony, which is like kingdom on the clubs bbs, but in basic and can be updated.

Most of the year when I typed out this articles I kept asking for people to get involved and so far it seems more people are and thats great, but remember we never have enough so if you would like to do a demo let us know and we'll set it up or if theres a demo you like to see let us know too. For a few months I will be setting up a demo on ams (advance music system), I will show how easy it is to type in music from song sheets to ams, so it can be listened to on the computer or if you have a 800xl with a gumby chip, it will play on that too.

Anyway I hope everyone had a great holiday as I did.

President Of NWPAC
Charles Lake

Y
O
U

This blank area is dedicated to those that have never sent in an article or had a question. We need your input so that next month this is not an empty space filled with trivia.

TO IOCB OR NOT TO IOCB

BY : MARK GRICE

January 1984, Vol 2, #4
Submitted By Jeff Rehder

Input/Output Control Blocks, commonly referred to as IOCB's, are probably the most versatile utilities the programmer has for extracting and storing information. Quite a few novice programmers shy away from them because they seem to be cumbersome and hard to understand, but they're really quite simple.

IT'S CHILD'S PLAY

When I think about IOCB's, I often recall a TV show I used to watch as a kid. It was called "Davy and Goliath." The main character, Davy, often communicated with his friends by using two Campbell's soup cans that were tied together with a string. I don't know if you've ever tried this method of communication, but I have. And in the process, I learned a number interesting things about it:

1) Before you can talk, you need to give an end to each of the parties involved.

2) Only one person can talk at a time. Because of this, you must decide whether you will talk or listen before the communication starts.

3) Before someone else can use the cans to talk, the first two users must end their communication.

These may not appear to be startling revelations, but if you understand these three simple rules you will have no trouble using IOCB's.

THE ELEMENTS OF AN IOCB

An IOCB always starts with an OPEN command, and looks something like this: OPEN #2,8,0,P:. The elements that make up this sample command work as follows:

* OPEN #2, - This sets up a line of communication (think of it as one of the cans, along with the string) that originates in the computer. This is called line #2.

* 8, - This tells us that we will be outputing (or, as in the tin-can example, talking). A four means we are inputing; a six means that we're inputing a disk directory; a nine means that we're appending a file that's already in existance; and a twelve can mean that we're either inputing or outputing.

* 0, - This tells us nothing, in most cases. It's just there. I know alot of you are screaming, "Why?" but I'm going to hedge on this anyway. Suffice it to say that there is a purpose for this part of the command, but that it's used so seldom for this purpose that it's not worth going into. Over 98 percent of the time, this number will be zero.

* P: - This is where the IOCB stops. It designates the other end of the IOCB channel -- who gets to use the can at the opposite end of the string, in a manner of speaking. (In disk operation, we'd also need a filename here, so the command would look something like this: OPEN #2,8,0,"D:FILENAME". In disk I/O, by the way, opening with the number eight will wipe out whatever was in the file previously, while opening with the number nine will cause the program to start writing on the next available sector on the disk, regardless of how much room is left on the previous sector.)

A LINE OF COMMUNICATION

What we have now is a line of communication from the computer to the printer (for the purposes of this article, we'll assume that a printer is being used for output). The computer is doing the talking; the printer is listening. A short sbroutine using this line might look like this:

```
20 PRINT "ENTER THE
   STUDENTS SCORE"
30 INPUT SCORE
40 PRINT "THE STUDENTS
   SCORE IS ";SCORE
50 OPEN #2,8,0,"P:"
60 PRINT #2;SCORE
70 END
```

Line 50 has already been explained. Lines 20 and 30 are pretty straight forward, and I don't think they require any further explaination. In line 40, we print the score in the regular fashion, to the screen. Line 60, however, prints the score to ICOB #2. This time, the information doesn't go to the screen - it goes to the printer instead. Why? Because we've told the computer to PRINT #2.

Going back to our example of the tin cans and the string, the scenario would look line this:

Davy is holding the tin can. On the other end of the string is his friend Bob. With Davy is another freind, Jane. She is planning a picnic, and wants to invite Bob. So jane asks Davy:

"What time should we hold the picnic?" (20 PRINT "WHAT TIME IS THE PICNIC?")

Davy thinks about it for a moment (30 INPUT ANSWER)

"Oh, about 3:30," he says. (40 PRINT " OH, ABOUT, ";ANSWER)

Then Jane says, "Better tell Bob," So, Davy picks up the can (50 OPEN #2,8,0,"P:") and says "Hey, Bob, the picnic is going to be at 3:30." (60 PRINT #2;ANSWER)

I hope I haven't insulted anyone's intelligence with my examples, but honestly, that's all there is to it! Anything connected to your computer can have an IOCB attached to it. This includes printers, cassettes, disk drives, the keyboard, or even a TV (Ever hear of a light pen?)

A FEW QUICK RULES

You can number your IOCB's using any number between one and seven (i.e., line #1, line #2, line #3), but you should avoid using the number 6, because it is sometimes used by the Operating System (OS). For example, have you ever seen a line like the following?

```
10 GR.2:PRINT #6;"HELLO"
```

In this example, the OS opens an IOCB to the graphics screen rather than the text screen.

When your finished with an IOCB close it. An END statement will close all IOCB'S. This ensures that all your transmissions will be received by the output device (in technical terms, it empties the buffer). If you don't empty your buffer there is an excellent chance that some of your data will be lost. To close an IOCB without ending the program, use the CLOSE statement (e.g. CLOSE #2).

Finally, after an IOCB is closed you can open it again to another device but you cannot have the same IOCB opened to two devices simultaneously (just as Davy couldn't have had one string attached to three cans) For more information on IOCB's, read your Reference Manual.

A FINAL EXAMPLE

Here's one final example of how you can use IOCB's effectively. The situation is as follows: you have to create a menu program that is to be used by someone that is not to terribly bright. There are two things he just can't seem to remember:

(1) That when you ask for a number, he can't enter a letter;
(2) Where the RETURN key is located on the keyboard.

The first program you submitted looked like this:

```
10 ? "PLEASE MAKE SELECTION"
20 ? "1) CHOOSE ONE"
30 ? "2) CHOOSE TWO"
40 ? "3) CHOOSE THREE"
50 ? "ENTER NUMBER PLEASE"
60 INPUT CHOICE
70 IF CHOICE=1 THEN
   GOTO 100
80 IF CHOICE=2 THEN
   GOTO 200
90 IF CHOICE=3 THEN
   GOTO 300
```

However, you quickly discovered that this would never work, because every time your user ran the program, he either started a letter or forgot to hit RETURN. An IOCB could help you tremendously:

```
10 ? "PLEASE MAKE SELECTION"
20 ? "1) CHOOSE ONE"
30 ? "2) CHOOSE TWO"
40 ? "3) CHOOSE THREE"
50 ? "ENTER NUMBER PLEASE"
60 OPEN #1,4,0,"K:":
   GET #1,CHOICE
70 IF CHOICE=49 THEN
   GOTO 100
80 IF CHOICE=50 THEN
   GOTO 200
90 IF CHOICE=51 THEN
   GOTO 300
95 GOTO 10
```

This method works like a charm! As soon as the user touches a key, the program takes off. No matter what the user does, he can only select a number. The explanation for this magic is that in line 60 we opened up an IOCB for input (because the second number is a four) from the keyboard. The computer then treats this input in the same way that it treats any input: it simply waits until a key is pressed

However, unlike the standard INPUT command, which must wait for a press of RETURN to tell it that an option has been selected, the GET command in this program takes effect after a single byte of information has been received. In this case, that single byte is always the ATASCII value of whatever key has been pressed by our hypothetical user.

By the way, using the IOCB GET command is not the same as PEEKing location 764, which I discussed in an earlier article (ANTIC, "I/O and You," page 16, July 1983). Location 764 gives you the interrupt value, which is the value you get before the OS gets its hot little hands on it. The IOCB GET command gets the value only after it is sent to the OS. This is why, in this case, the IOCB GET command returns the ATASCII value of any key that is pressed.

As a result, when we use this method we know for certain that we will end up with a number, rather than a letter, no matter what happens. So, we can say goodby to Error 8. Incidentally, if we want to print the selection that our user has

made, we can do so easily by adding the following line:

```
65 PRINT CHR$(CHOICE)
```

In conclusion, if you want constant input that you can "grab on the fly," use location 764. If, on the other hand, you want the computer to wait for it, use an IOCB. Only after exhausting these two possibilities should you use an INPUT statement! Try them. I think you'll find that these two options will help make your programs much more "user-friendly."

* MEETING SCHEDULE *

NWPACmeetings are held on the second Saturday of every month at the:

FAITH UNITED METHODIST CHURCH
8648 N. 19th Ave.
Phoenix, Arizona

09:00 - 09:30 Informal Talk
09:30 - 12:00 Meeting

Meetings are open to the public. Visitors may attend 3 meetings, before being required to pay membership fees.

Single Membership: $15.00/yr
Family Membership: $15.00/yr
Affiliate Membership: $10.00/yr

Don't wait to Join

There is an additional charge of $2.00 if you wish to have the Newsletter sent to your home. This is to try and cover some of the charges the club incurs. If you want the newsletter mailed to your home. All you have to do is send in $2.00 to:

NWPAC Newsletter Mailing
P.O. Box 36364
Phoenix, AZ. 85067

Yes, this is the same address that we use for our mailing, but it's helpful to know what the letter is about, so it can be given to the proper officer. We Thank you for helping to keep the Atari 8-Bit alive and kivking...

25

PLEASE RUSH

DATED MATERIAL

EDITORIAL

I would like to take this opportunity to introduce myself as the new newsletter editor. I have a degree in computer sceince and have had an Atari computer since 1982 I jioned the club on a family membership with my father 3 years ago in Feburary and have since gotten my own menbership to have more opportunity to give and gain knowledge within NWPAC.
The newsletter is administered by the Executive Officers with the exception of the editorial column. Items reported here may be controversial informative, educational, but they will be my personal observations.
This month I would like to discuss courtesy or in this case lack of courtesy. Almost every meeting I have attended you can hear people talking and others walking around or actually leaving during the demonstrations unfortunately club officers are the primary offenders. These are a few suggestions that might help. No disk sales, except prior to the meeting and between demonstrations. Club officers should turn the meeting over to whom ever is giving the demo and have a seat. They may even learn something new. Courtasy is contagious, it spreads from person to person and meeting to meeting. So lets start this meeting by listening and showing respect to the person at the front.
I hope that I can add to the newsletter to make it for you with what you want in it. If you have any suggestions please leave me E-Mail or call and we will do our best to serve you.

Editor -N- Chief
Mike Barnes

PD DISK COST 1991

Series	Cost	Per	Bonus
100-499	2.00	DSK	3 get 1 free
500-5099	3.00	DSK	3 get 1 free
Analogs	4.00	DSK	
Adult	3.00	DSK	
Catalog	4.50	BOOK	
Cat updates	.03	SHEET	
Blank disk	10.00	PKG)25)	3 get 1 free

Purchases can be made at Club meeting or at Librarians residence with Advance arrangements.

Librarians home phone is 894-9156 if you need to make the arrangements.

North West Phoenix Atari Connection

Here's your Feburary supplement to the Nybbles & Bytes..

At the last meeting, Charles announced the new NWPAC officers. The meeting started at about 9:40, and John Collins was first up to bat. Announced, the new arrival of an international network, for PAUGS. Kevin (Iron Horse's SysOp) has gratefully started pulling down messages from California, at his expense. Nancy Gibson brought in a catalog from DataRush in Ca., featuring many good deals in software.
Later Carl started a discussion about the possibility of changing the Constitution. The Pro's and Con's were flying, but after 45 minutes of it, it was decided to set it aside, to a later date.
After our 30 Minute break, Charles reconviened the meeting, Topic was a New meeting place. The Club has untill March to make a decision, and places were suggested.

After all this, we still didn't have access to the churchs' monitor, so we broke into a Q&A session. At that point, members would ask about different problems they were encountering, and a solution was sought.

Dave Farley

Treasury Report

By : John Collins

I would like to take this moment to introduce myself as your new Treasurer, and also to apologize for the lack of a report in the last N&B. At that time, I had not received our accounts from the outgoing Treasurer.
As for myself, my background includes experience in Accounts Receivable, while working as a full-time employee at UCLA. I've had an Atari since 1985, and am primarily interested in Graphics.

On with the report, as you can see we were practically dormant in the month of December, but made up for it with a flurry of transactions in Janurary.

December 1990 finances:

Beginning Blalnce		$ 625.07
Income -	Membership	65.00
	Disk/Raffle	116.50
Expenses	NewsStation	(63.95)
Ending Balance		$742.62

Janurary 1991 finances:

Beginning Balance		$ 742.62
Income -	Membership	30.00
	Disk/Raffle	130.00
	Misc	.01
Expenses	Printer	(149.27)
	Phone	(46.95)
	Stamps	(25.00)
	Printing	(21.95)
	Ribbons (6)	(16.56)
Ending Balance		$642.90

Membership

Attendance	23 Members
	3 Visitors
	2 New Members
	1 Renewal

Welcome Ted Rabino & Gary Northcutt

Winner of the Raffle:

HERB CRAFT (AGAIN)
Herb chose Swat/Panther

Coming Attractions

Coming in Feburary:

Basic With Paul Parks
Zybek by Carl

Coming in March:

Basic with Paul Parks
Rambo 256K by Danny Hayes

The following members are due for renewal, if your name is here as expiring in Janurary, this will be the last supplement you receive.

Name	Due Date
Dennis Zismann	Jan '91
Merle Barnes	Feb '91
Jerome Bruschette	Feb '91
Grant Holcomb	Feb '91
Charles Lake	Feb '91
Michael O'Neil	Feb '91
Paul Parks	Feb '91
Don Clegg	Mar '91

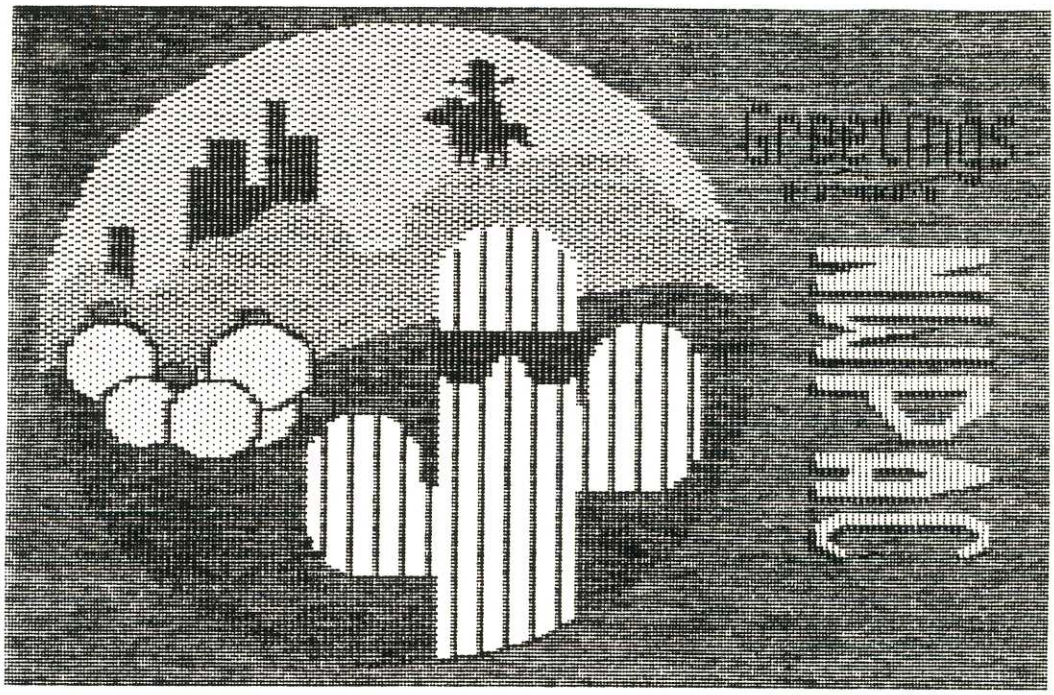
I would have added names from April, but there aren't any..

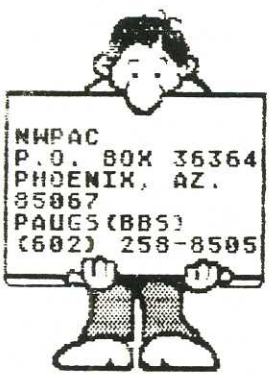
PD DISK COST 1991

Series	Cost	Per	Bonus
100-499	2.00	DSK	3 get 1 free
500-5099	3.00	DSK	3 get 1 free
Analogs	4.00	DSK	
Adult	3.00	DSK	
Catalog	4.50	BOOK	
Cat updates	.03	SHEET	
Blank disk	10.00	PKG)25)	3 get 1 free

Purchases can be made at Club meeting or at Librarians residence with Advance arrangements.

Librarians home phone is if you need to make the arrangements.


Greetings


NWPAC
P.O. BOX 36364
PHOENIX, AZ.
85067
PAUGS(BBS)
(602) 258-8505

TO: