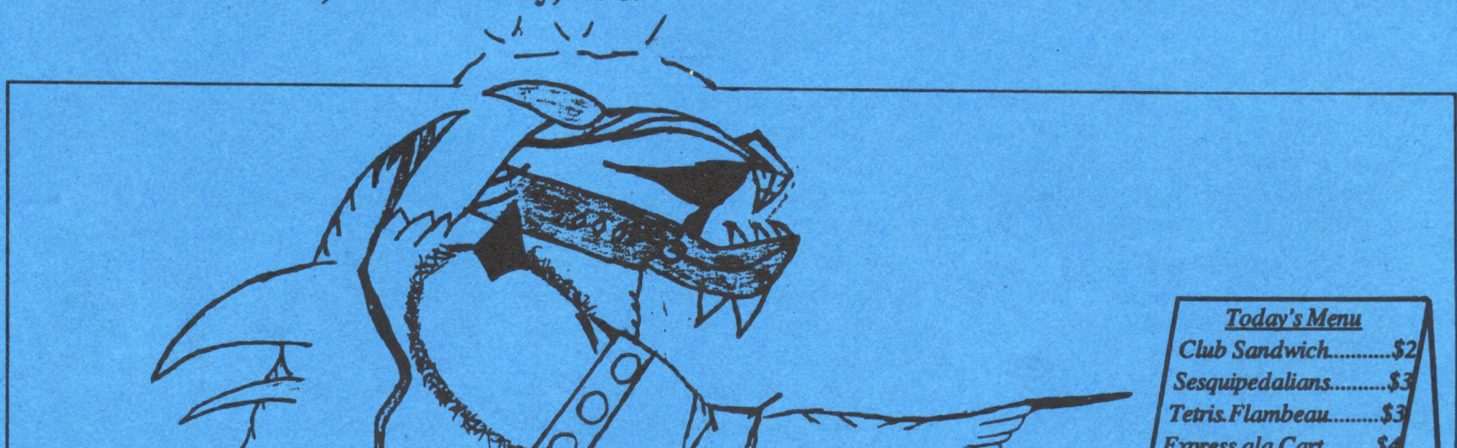
DAL-ACE

Dallas Atari Computer Enthusiasts

Volume 10, Issue 7 July, 1989


Today's Menu
Club Sandwich...........$2
Sesquipedalians..........$3
Tetris.Flambeau.........$3
Express ala Cart.........$4
Portfolio Lunch..........$6
Stacy's Goulash.........$6
Broken Dishes............$8
Format11 Surprise.....$8
Power Chips..............$8
Boulderdash.Hash..$10
Forth.Turn Overs.....$11
Sam! That Ape-bit looks hungry! Maybe you better feed him!

MEETING DATES: JULY 8, AUGUST 12.

Thank you Ralph

Sounds like we're in a rut since the last month's newsletter started off with "Thank you Angela". This is a rut we would rather not be in.

Ralph Salmeron has announced his imminent departure for a new job in Ann Arbor, Michigan.

Ralph is one of the many people, who labor mightily behind the scenes, doing the necessary tasks that keep the club running. We rarely notice these people until they are gone, and the services they provide are no longer there.

Among the many jobs that Ralph has been doing is that of Sysop for the club BBS. He is also an elected member of the Board of Directors, and has been very active in gathering up new material for the club's 8-bit disk library.

Ralph, we will miss you greatly, but we want to wish you the best of everything in your new job.

Dollar $igns

by Rene Tucker, treasurer

For those of you who were not present at the meeting on June 10, 1989, here are the highlights of the financial report for the month of May.

RECEIPTS	$760.25
EXPENSES	$429.35
GAIN	$330.90

Remember, a full report is made at each regular meeting and copies of the monthly reports are available for inspection by any club member.

Highlights of the June Meeting

For those of you who were unable to attend the June meeting, here is what you missed.

Not much unfortunately. It was more of a case of us missing you.

June is, apparently, a big vacation month that is coupled with many other once-in-a-lifetime must events, like graduation ceremonies, out-of-town guests, and unavoidable work assignments.

June fever had apparently affected many of the club officers as well, since many of them were missing at this meeting.

Main Meeting

The club president and the vice president were among the missing, so, the main meeting was called to order by the VP of Communications, Anita Uhl.

Anita did a bang-up job of running the show in spite of the short time she had to prepare for the meeting.

Treasurer Rene Tucker presented her financial report for the preceding month. There was a substantial increase in the club bank balance as a direct result of the very successful auction held last month. Rene expressed the club's appreciation for all those who contributed to the auction in the form of items to sell, and money to buy. Also, there was a substantial expenditure reported last month due to the purchase of additional blank disks for both club libraries.

Anita announced that Ralph Salmeron would be moving out of state, and asked for a volunteer to take over as BBS Sysop. Jim Jackson eagerly raised his hand. Anita fainted.

Not really, but after recovering from the shock, and since the members were in general agreement, Jim Jackson was nominated to be the new BBS sysop. Congratulations Jim.

Once again, Anita asked for volunteers for the position of Newsletter Editor. No takers.

Come on folks, typing in thousands of words each month is a fun job, providing you have more than two fingers, and a degree in English, (which the current slave has neither).

Actually lots of folks have helped out this month by contributing articles and typing up the reprints, so the slave, bows and scrapes, and says thank you all very much for your help.

During the question and answer session, Ralph Tenny volunteered to assist anyone in need of PC-Ditto help. Go git him folks! We need to keep these enthusiastic volunteers enthusiastic.

The same goes for Jim Jackson. Call up the bulletin board and pester Jim.

After a sterling performance as Master of Ceremonies, (in spite of a knocking sound coming from under the table), Anita adjourned the meeting and we went on to disk sales, member garage sales, Marc Salas's New Member SIG, and the ST SIG which provided some excellent demos by Ralph Tenny on a real live ST.

The garage sale featured the unloading of all of the 8-bit property of an old time club member, Harry Hafele, now into IBM PCs, but once the Editor of this rag.

Harry taught me Forth, so if you have been following the Forth articles, I want you to know that Harry's legacy to the club lingers on.

All in all it was a really neat meeting in spite of all the missing vacationers.

ST SIG Meeting

by Jim Jackson

Thanks to Ralph Tenny we had a very good demo of Quick Draw at the June Dal-Ace club meeting.

Ralph demoed Quick Draw as a program he uses when designing various electronic circuits. He brought with him a few copies of Modern Electronics, in which some of the diagrams he has drawn have been published, to show the camera readiness of the finished product.

Ralph showed the various "built-in" symbols and how they could be edited into one's own symbols. He also showed how to move symbols, incorporate them into schematics, and create line drawings.

Mr. Tenny, after completion of the Quick Draw demo, did a comparison with Circuit Breaker.

The comparison between the two programs showed both were powerful tools in his line of work, and each had its own points above the other.

At the end of the demo however, I came away with the feeling that Quick Draw was the better of the two and would be my recommendation to anyone in need of an electronics designing tool.

8-bit SIG

The 8-bit SIG met at 10 am. Thanks to Ralph Salmeron, we had two new club disks to show. Analog #73 and Dalace #110.

The Analog disk features AGENT.OBJ, an adventure game, ALPHABET.BAS, a utility that sorts a disk directory and rewrites it in alpha sequence, SEC2PRNT-.BAS, a program that tries to create dungeon maps for your adventure games, and MARBLEMG.BAS, a jump-the-peg game.

Dalace disk #110 features COPYMATE 5.0, a super sector-copier that makes short work of copying any non-protected disk, including the fig-FORTH disks.

Also demoed was the July disk from Antic. RED SQUARES is a basic program that emulates the Russian-written program TETRIS which is available commercially for several machines other than the 8-bit, (see the review of TETRIS ST in this issue).

Antic programs are not public domain, so if you want to play RED SQUARES give Antic a buzz on (800) 234-7001. Ask for the July 1989 disk. The price is $5, but it is worth a lot more.

The 8-bit SIG ended on that note, with several members continuing to play RED SQUARES on into the afternoon.

TALK

from Randy Randolph

The editors of this news letter have been accused of being sesquipedalians. To that end we feel it is necessary to make a statement of the editorial policy that governs this institution.

Hence forth, it shall be Dal-Ace policy that All rules and regulations, communications and inquires, reports and releases and in general, all club business and activity be devolved in the most basic English suitable to the purpose. In particular, it should be emphasized that a large portion of the interface coordination communication must now utilize, and certainly be functionally interwoven with, wordage of a non-sophisticated nature.

Glossology aside, a constant flow of effective information tends to maximize the probability of project/activity success and minimizes the time and cost required for implementation of subsystem compatibility archival. Therefore, a fully integrated language/word system must be used with respect to specific goals and a primary interrelation between word technologies and the end rational be primarily considered. Certainly, we have reached the point of satisdiction on this subject, the position is clear.

(Adapted from John Gill's address to the Council, Chapt 4, Verse 3)

TETRIS The Soviet Challenge

by Donny Arnold

Tetris is one of the most challenging games I've ever played.

You have six different shaped pieces that fall from the top of your screen. You may turn, move left or move right. Once in place you may drop the piece to increase your score.

The object of the game is to align all the pieces so that there are no gaps in the layers across the bottom of the screen. As you make a full line across the screen it disappears. After a certain number of lines disappear, the square pieces fall at a faster rate.

There are ten levels that must be completed. As the screen fills due to uncompleted lines, you have less and less time to align the pieces. The tension builds. It continues to build until you just throw yourself back and yell **"I JUST CAN'T TAKE IT ANY MORE!!!"**.

My highest score so far is 5965 at level 9 and three tranquilizers. What a game. You have to try it. Just don't blame me if your computer ends up needing repairs.

ATARI 8-BIT TELECOMMUNICATIONS COMES OF AGE!

The New Express! (The Cartridge)

by Farley D. Barge
Editor G.R.A.S.P. GAZETTE

This article may be freely distributed provided, proper credit is given to THE GREATER RICHMOND ATARI SUPPORT PROGRAM, the author, this notice remains intact in your final copy and the contents are not altered for any reason other than grammatical errors or spelling.

I have seen the future of Atari 8-bit modeming and it's Express (The Cartridge) by Orion Micro Systems! Yes the Express cartridge is nearing completion and on Thursday March 23rd at 7:00 p.m. Chris King and Keith Ledbetter demonstrated a working cartridge to a full house at the GRASP meeting. To say that it was impressive is an understatement.

In case you aren't aware, Keith Ledbetter wrote the classic 8-bit terminal program 1030/850 Express nearly -- years ago. It has remained the terminal program of choice for most Atari 8-bit users and is the one virtually all others are measured by. If you haven't ever seen Express, check out your local Atari club library or BBS and your sure to find a copy. In addition to being the finest terminal program available for the 8-bit it's also in the public domain!

Well all the above was true up until now, because the boys from Midlothian have really outdone themselves this time with Express (The Cartridge). In case you haven't heard here's a rundown of a few of the features packed into the cartridge. Supports most Atari Dos's including the SpartaDOS X cart. Intuitive drop-down menu system. Support for 130XE and upgraded 800XL's expanded memory up to 528K. Access to most common DOS commands, ie. erase, rename, dir, type etc. Full screen text editor Works with any modem using 'R:' handler. Compatible with Atari 800, 800XL, 130XE and XE game machine. Supports 300/1200/2400/4800/9600 baud rates. New high-speed screen handler means no more lost characters while capturing text. Supports R1: thru R4: comm ports Supports the most popular transfer protocol. Runs external disk-based programs. Exit to DOS without loss of carrier

Whew! Ok, so does it really do all of those things? The answer is an emphatic YES! The Express Cart is based on the super cart technology developed by ICD. In fact the production model of the cartridge will actually be burned and silk screened by ICD. Distribution will be handled by Orion Micro Systems.

The cart is 64K of bank memory written in 100% assembler code. The drop down menus and exploding windows are very impressive and much quicker than those in GOE. However, the most impressive thing about Express Cart is the high speed screen handler.

Our demonstration was set-up between two 8-bit Atari's, one running BBS Express Professional and the other running Express (The Cartridge), both connected via a null modem cable communicating at 4800 baud (we had to slow it down so every one could read the text). The Express Cart was piggy backed with a Sparta DOS X cart.

Keith has completely rewritten the screen handler to produce extremely fast text scrolling. This was necessary so that the program could keep up with the higher baud rates commonly in use today ie., 2400 and up.

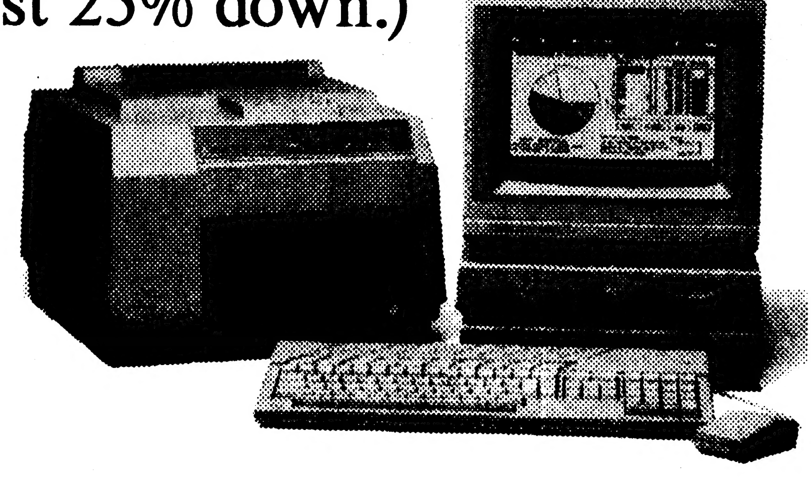
Chris gave us an over view of the features and a brief history of the evolutions of Express from 1030 Express, 850 Express up to the new Express Cartridge. At that point Keith jumped in to run us through most of the features listed above.

One interesting point about the new program is that it has been totally rewritten to allow execution of external modules. For instance, if some one creates a new whiz bang download protocol, a disk based handler can be written and accessed by Express Cart to handle it. This is possible because Express Cart, just like it's cousin BBS Express Professional, consist of a shell that loads smaller program modules to handle various task performed. This allows Express Cart to continue to grow as long as there is a programmer out there willing to write modules to fulfill the growing needs of us telecommunicators! There's a lot of horse power in this little 64K cartridge!

Currently Orion plans to have completed beta testing and begin deliveries by approximately the 1st of June, 1989. Judging by the product we saw (which was a fully working program burned into a cartridge) they shouldn't be to far off of that target.

The price of Express (The Cartridge) is $69.95. If you order before June 1, 1989 you can receive a $10.00 dollar discount bringing your cost down to $59.95. For more information contact:

Orion Micro Systems
Attn:Express! Cartridge
2211 Planters Row Drive
Midlothian, Virginia 23113



COMDEX/SPRING '89

At Comdex this Spring, Atari was showing two new products.

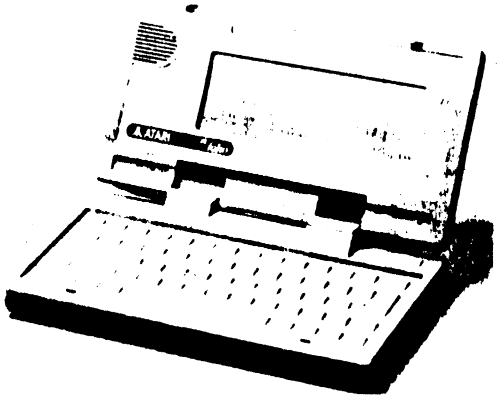
THE PORTFOLIO

Atari has introduced the first **handheld** IBM(R)-compatible personal computer. The 7.8 x 4.1 x 1.2 inch system, (about the size of a video cassette), has a 63 key IBM compatible keyboard and offers an 8-line by 40-character LCD display. It is designed around the Intel 80C88 microprocessor, weighs 1 pound and operates on AA batteries. The batteries are expected to last about a month.

This system is MS-DOS (R) 2.11- compatible and has 128K of RAM as standard (expandable to 640k). Disk drives are replaced with credit-card size 32K or 128K RAM cards. A port is also included for file exchange with desktop or laptop PCs.

The Portfolio is provided with a calendar program, address/phone list card, text processor, spreadsheet.

The Portfolio handheld computer will be available for delivery in June at a suggested retail price of less than $400.



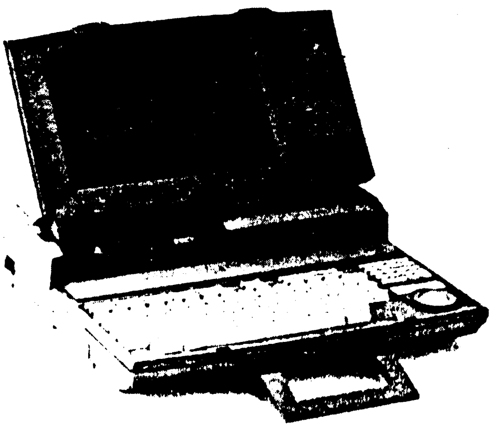
Atari's PC Folio

STACY

This laptop system is an addition to the ST line. It features a megabyte of on-board memory, a 3.5 inch doublesided floppy disk drive, an LCD monochrome display with 640x400 resolution.

It has all of the standard ST computer ports and interfaces, a fully-functional St-compatible keyboard with 10 programmable function keys and a built-in trackball for use as the mouse control.

The Stacy laptop has a suggested list price of $1,495 and will be available for national distribution in June.



SubLOGIC Drops Atari Support

Reprinted from ACORN, June 1989 issue. Author unknown.

ACORN member Pat Baratta recently engaged in an exchange of letters with SubLogic Corporation regarding the inability of Flight Simulator II to run on his Indus and XF551 drives. Their replies follow:

From Tammy Shafer, Product Support Technician on August 25, 1988:

"If your disk drive is an INDUS GT, there can be complications. Flight Simulator II and other SubLOGIC programs were developed using all ATARI hardware and disk drives. The INDUS is not completely 100% compatible with our programs.

"The Atari Flight Simulator II was written using the 810 and 1050 drive. If either the ATARI 810 or 1050 disk drives are used, there should be no loading errors unless the disk somehow became defective. We have found that the new Atari XF551 drive is not compatible with the AT-FS2. This is something that will possibly be changed in a future update."

From Jeri L. Rose, Sales and Customer Service on May 19, 1989:

"The problem with the XF551 disk drive would have to be solved by Atari. Since Atari is not going to correct the incompatibility between our products and the disk drive, we have discontinued all of our Atari line products. We know there are many Atari users who appreciate our products, however it would be financially immoral to continue our products when more and more people are using a disk drive that will not run with our products. We appreciate your concern, and I wish there was something I could do to help in the situation, but I can't. If you have any further questions or problems, please contact me at POB 4019, Champaign, IL 61820 or call 800-637-4983."

Sof-Tek Marketing

5601 Ammons, Haltom City, TX 76117
METRO 817-498-1924

Airborn Ranger	32.50
Art & Film Director	52.50
Battlechess	31.00
Blasteroids	26.50
Calamus	179.95
Cosmic Pirates	27.50
Cyber Texture	32.00
DB Man 5.0	159.95
Desktop Publisher ST	84.50
Drive Master	34.50
Emannuelle	26.50
F-16 Combat Pilot	34.00
F40 Pursuit	30.00
Falcon ST	31.00
Flair Paint	49.95
G+Plus	24.50
GFA Basic 3.0	64.00
Hisoft Basic Prof.	105.00
Hot Wire	25.50
Hyperfont	32.25
I-Ludicrus	24.50
IB 5 1/4 Disk Drive	205.00
IMG Scan	65.50
KCS 1.6	149.95
LDW Power	89.95
Laser C	115.00
MFD 1080 3.5 - 5 1/4	319.95
Mark Wiliams C	100.00

Midi Max	32.50
Mouse Master	29.75
Multidesk	20.50
Neo-Desk II	31.50
Operation Wolf	27.50
Page Stream	124.00
Populous	34.00
Precious Metal	34.00
Prison	25.50
Pro Copy ST	23.00
R-Type	27.95
Revolver	32.50
Robo-Cop	29.95
STOS Basic	43.00
STOS Compiler	23.75
Sargon III	34.50
Spectre 128	124.50
Spectrum 512	46.00
Tele-Pak	154.00
Touch Up!	124.50
Tweety Board	40.00
Ultra Script	149.95
Video Key	72.00
Vindicators	27.25
War in middle Earth	32.00
Zak Mckracken	32.75
Zany Golf	29.00
Zoomer Flight Yoke	44.00

BATTLECHESS
3D Animated chess with incredible animated players!
$31.00

HOTWIRE!
Run any ST program by simply pressing a hot-key from your keyboard. A great value for hard-drive users!
$25.50

F-16 COMBAT PILOT
Better than Falcon? Reviews say it has better missions, but it looks like Falcon has the edge on graphics.
$34.00

ULTRA-SCRIPT
Turn your dot-matrix or laser printer into a post-script compatible device. Great for desktop-publishers!!
$149.95

THE BEST ATARI ST SPECIFIC MAGAZINES!
ST-USER $4.95
ST-ACTION $4.95
ST-WORLD $4.95 (European)
STart MAGAZINE $2.95
If you order over $100.00 worth of products from our company in one month we thank you with one of the above magazines...FREE!!

Products to watch for..
PC Ditto II (low...)
Seymore Raydix VCR Back up (low..)
CMI Processor Accelerator (call...)
Chaos Strikes Back!(18.95)

Hot European software due out soon...
Altered Beast, Shinobi
Power Drift, Ferarri Formula one
Buy those European MAGS!!

Sof-Tek Choice PD arriving soon...

Our goal is to provide you with the best selection of ST products at the fairest pricing possible. We understand that money is not easy to come by, therefor if you see any of our products advertised lower elsewhere we'll do our best to match or beat thier price. Please keep in mind we are not a retail store. (yet..) Please call first before coming out to our location. We offer UPS shipping and free delliverery in person to Dallas & Ft.Worth. Please call to see if your location qualifies for free delivery. We would like to put you on our mailing list. Our catalogs feature over 200 items, with news on monthly raffles and special offers. Sof-Tek Marketing currently is only supporting the Atari ST.

RESTORING COPY PROTECTED DISKS

by Jeff Golden

We have had considerable success in restoring crashed floppy disks. What you need are a sector editor and a friend with the same version of the program. (Sometimes, if you have more than one drive, you may be able to find a drive that can read the bad disk, eliminating the need for the friend's copy.)

Since the bad disk is essentially worthless, you have little to lose by following these procedures. The worst you can do is make the disk worse than worthless, if there is such a thing.

1. Use the sector editor to determine the bad sectors on the defective disk, and write down the sector numbers. A sector copier may be able to perform this portion of the task much faster.

2. Use the same procedure on your friend's disk after making sure that it is write protected. Your friend's disk will probably show some bad sectors as part of the copy protection scheme. Do not attempt to copy these sectors.

3. Compare the sector before each bad sector to see if you have the same version of the program on both disks. Then copy only the sectors that are bad on your disk, but good on your friend's disk.

Presto, your worthless disk may run again. Next time, keep it away from paper clips and other metal objects, and don't leave it in the disk-drive when you turn the drive on or off. The next flip of the switch may be the unlucky one.

THE TROUBLE WITH FORMAT11

by Chris Freemesser

Reprint from ACORN, June 1989

So you have your ST, but you are unhappy with only 720k per DS disk. You find a format program that will increase your disk capacity to over 800k. You think to yourself "Gee, this is GREAT!". But is it? Maybe not.

A program called Format11 by Matt Orsie has been floating around for some time. I used this program on all my disks, be them SS or DS. Then I started having problems with my disks. Directories would not change when new disks were put in, garbled directories would pop up, and sometimes the computer crashed. At first, I thought my drive was going bad.

It was not the drive. It was the program! I found out that other people had the same problem, and the culprit was a logic error. Put simply, the ST assigns an identification number to each disk when it is formatted. The ST uses this (in part) to determine when a new disk is inserted. Format11 put the same number on EACH disk it formatted. The computer would get confused when it saw the same number on a new disk. I ended up spending an entire weekend copying my disks over to ones with a GOOD format on it.

So the warning is: DON'T USE FORMAT11! If you want to get more capacity out of your disks (and there is nothing wrong with this), I recommend DCFormat from Double Click Software. It is a SHAREWARE program, but is well worth it.

CHIPS/POWER

by Chip Scoppa

Reprinted from ACORN, June 1989

This column will appear from time to time in the Kernel and will deal with the use of LDW POWER (Logical Design Works' spreadsheet program).

First a little about Chip. I am an engineer and use PC's commonly in my everyday work. Up until a few months ago I knew very little about ATARI and even less about the ST, but I was fortunate enough to meet a few people who were committed to the ATARI way of computing.

After a couple of days at the new job here in the Rochester area I had scared up enough parts to put a PC on my desk without having to ask for the company to purchase a new one. I was pleased with the system that I had working and with the spreadsheet program that was on its hard drive.

The program was one that I was very familiar with and have enjoyed using. It is called LOTUS 1-2-3. (Now you can tell that the system which I was able to find was at least Big Blue compatible.)

Near to my office there were some unusual computers working and I got very interested when I started to see the kind of work that was being done using this new and strange kind of "PC". The work seemed to be of a better quality and seemed to be easier to do. I became more and more familiar with this new kind of "PC" and soon decided that I would like to have one of my own.

I found very few places locally to purchase a new Atari 1040ST, but the SwapSheet had an ad for a used system. This system now has a new home and resides in my family room.

A few pieces of software came with my "new" used system. One was a LOTUS 1-2-3 look alike called VIP Professional. I was pleased to find that the time that I had invested learning 1-2-3 was not wasted, although many parts of the program seem a little sluggish and slightly hard to use.

I saw a review in STart of another LOTUS 1-2-3 look alike for the Atari ST computers. Within a few weeks I had purchased my own copy of this program called LDW POWER by Logical Design Works.

Not only is LDW POWER a Lotus 1-2-3 look alike, but it will handle 1-2-3 files and convert them to its own type of file. LDW POWER can also convert its files to a form that 1-2-3 can use.

LDW POWER can retrieve a file that has been saved using itself or files that have been saved using Lotus 1-2-3. Here is how to save and retrieve a file using LDW POWER.

1. Once you have turned on your computer and loaded the LDW POWER program use the mouse to point to FILES or type in the command "/F" (anything that is to be typed in I will enclose in quotes). Now you will see either the drop down menu or a changed command line depending on the way you got to FILE sub-menu. The 'drop down menu' will appear when you use the mouse and the changed 'command line' will appear when you typed in your command.

2. Next point to 'RETRIEVE' or type in "R" and the file selection box will appear with all the files on your disk in the directory selected in your LDW POWER program. Note that only the files with the LDW extension are shown. Point to and click on the file that you want to load into the spreadsheet framework. You can double click on the file or single click on the file and single click on the 'OK box' to load the file. This will allow you to load any LDW POWER file into your program that is in the directory on the disk in the drive which is set-up within your program. Later columns will cover how to change and save the internal setups within the LDW POWER program.

3. Once you have worked with this retrieved file or a new file and want to save it, you go through many of the same steps as you did to retrieve the file. To save the program point to 'FILE' and point to 'SAVE' and click or type in "/FS" the file selection box will appear with name of the spreadsheet on the select line. If this is the name you want to save this file with just click on 'OK' or press "Return". If you want to change the name or the save path do it just like explained below.

4. You can expand the listing in the 'file selection box' by pointing to, clicking on and changing the line with the directory and the file retrieval path listed. The path will look something like this: A:\*.LDW. The A: means that you are going to retrieve your program from the A drive. The \ means that you are looking to the ROOT (or the so called main) directory on the disk in the drive. The * means that you are looking for any file in that directory and the .LDW means you are looking for only the files with the LDW extension on them. When you point and click on the directory line a small up and down line will appear at the end of the line. You can move this line using the left and right arrow keys and the backspace key. If you backspace into the line and delete the LDW, but leave the . and then click on the upper left corner box (DO NOT press "Return") you will see a * appear in place of the LDW. The file selection box will fill with all the files on your disk from the directory that has been previously selected. You can change the directory you want to look into or the drive you want the files to come in from in the same way. Just point, click, move, change, and click on the upper left box. Now you can with this expanded list retrieve a Lotus 1-2-3 file if it is in your directory on your disk that is in the drive that you chose.

5. Now to change a LDW POWER file to a file which LOTUS 1-2-3 can use you need to run a program on your LDW POWER program disk in the LDWCNVRT sub-directory. Run the program called LDWCNVRT. It will ask you to choose the LDW extension file to be converted to a LOTUS 1-2-3 type of file. Next it will ask you where to write the file. When you tell it the answers to these questions by changing the file selection boxes as explained above the program will do all necessary conversions to the file to make it compatible with the less POWERful program.

Thank you for reading this first column. If you have any questions about LDW POWER or its applications please let me know and I will try to answer them for you or put the answers into future columns. My address is: Chip Scoppa, 8 Chesfield Lookout, Fairport, NY 14450.

TWO SUPER BUYS

by Nicholas J. Cup

Reprinted from ACORN, June 1989

You may have noticed that it is getting harder and harder to find software for your 8 Bit computer. I'm not even talking about good software mind you, just software. So when you find two pieces of superb software that are also priced more than reasonably, you (I) feel the need to tell someone about it.

I have recently purchased Super Boulder Dash from Electronic Arts, and Boulder Dash Construction Set from Epyx. I don't understand how it came to be that two different companies sell programs that are based on one another. But no matter, I have them now, and I'm very happy!

The slightly stranger thing about these two programs is that they are both based on another program called (you guessed it) Boulder Dash. This is a fine program, that has GREAT playability, GREAT graphics, and aside from being very addictive, it is a lot of fun to play.

The basic premise of Boulder Dash is that you control a character called Rockford. He is a cute little guy with a craving for diamonds. You, (or is it he), go through different screens collecting them and avoiding the dangers of the caves. These caves not only test your reflexes, but more importantly, get you to do some very fun problem solving. Each of the caves are different, and present new problems to overcome. There are 80 caves in all.

You may start on every fourth cave if you would like. This is very helpful to practice the screens nearer the end. You may not jump to the last two levels though, you must earn your way to them.

Some of the obstacles you encounter are: Fireflies, Butterflies, Slime, Amoebas, and falling anything. The Fireflies blow up if hit by a falling rock, and blow you up if they touch you. Butterflies also blow up when a falling rock drops on them, but they also create diamonds. They will also blow up on contact with amoebas, and you also blow up if you are touched by them.

Amoebas grow on the screen and they can also grow at different rates of speed and lengths of time. If you can enclose it they turn into diamonds, if not, they become rocks.

Super Boulder Dash is every bit as good as Boulder Dash. In fact, it will really make you think. But thinking is fun, try it.

Now that you understand the basics, we can move on to Boulder Dash Construction Kit. This is also a very well thought out program that will allow you to create your own custom screen effortlessly.

I have enjoyed creating screens for hours on end. Of course the best part about creating them, is to share them with your friends (who own the program), and see if you can drive them crazy.

I paid $14.95 for Super Boulder Dash and $9.50 for Boulder Dash Construction Kit. I think that these are very good values and you should consider getting them.

There are some things I would like to mention before I end. One of the problems with each of the games is that it is sometimes difficult to push a boulder on one of the meanies. This at times can cause your death.

Another thing to mention is that Super Boulder Dash is copy protected and the Kit is not. The only problem I have with this is that since companies like Electronic Arts seem to forget to support the 8 Bits when ever they feel like it, what happens when the disk crashes? I can't back it up and probably can't replace it. So then I'll need a Boulder Dash fix, and have no place to get it.

On the other hand the Kit can be copied. So if you wanted to, you could give it to your friends. But if you want these companies to continue publishing great unprotected programs at fantastic prices, you should pay for your programs, not pirate them. Why? Because I want more great, reasonably priced, unprotected software for my floundering 8 Bit. Otherwise we will not see anymore new programs for our computer, nor will we be able to find any good old programs that were available last year.

So think about it the next time you steal a piece of software and you feel you haven't hurt anyone. Because if you just look around and read the writing on the wall, you will see that most companies that are selling Atari software are selling it to get rid of it and clear their inventory. The next step is DEATH!

Irving World of Atari??

Dave Munsie reports that St World Magazine, the sponsor of the World of Atari shows, has been running an ad for a show to be held in Irving in August...

FORTH TUTORIAL

by Jeff Golden

PREFACE

This is the first in a series of files that will make up a tutorial on the FORTH language for both the ST and the 8-bit Atari computers. Since the FORTH language is more or less standard across many computers, and since the object of this tutorial is to teach standard FORTH, then the tutorial should be applicable to many other computers as well.

We hope to cover the FORTH WORDS (functions) one by one as the tutorial progresses. Eventually we will get into writing some simple programs, but we must first learn to walk before we start running with the language.

Please feel free to enter any questions and comments about the FORTH language, or this tutorial, in the programming message base on this BBS. The author feels that the student questions are just as important to the learning process as anything he might have to say. It would be wise to also capture and print those questions and answers as we go along for future reference.

One more word about learning FORTH. HANDS-ON experience in using FORTH is very important to the learning process. After downloading and printing out each edition of this tutorial, please sit down and go through the exercises that we present, even though you feel that you thoroughly understand the subject matter. The importance of discovering that you can do it yourself cannot be emphasized too strongly. Seeing what happens when you make a mistake is also very educational.

INTRODUCTION

To avoid lengthy explanations every time we do something, let's use a coding system for the things that you should type in, i.e. pressing the carriage return key, etc.

5629 <cr> (indicates that you should type in the number 5629 and then press the return key)

5 6 2 9 <cr> (indicates that you should type four separate digits, separated by spaces, and then press the return key.)

Be very aware of the spaces inserted in the command strings as these spaces will act as delimiters, separating the values and words that we will be feeding to FORTH.

A FORTH WORD is defined as a sequence of from 1 to 30 characters followed by a space that has been previously defined to FORTH. Almost any character you can type from your keyboard, other than a space, can be used in a FORTH word. In a short while you will learn that the ! @ . (exclamation point, at-sign, and period) characters, when delimited with a space, are actually three single-character, standard FORTH words, and when encountered will perform specific FORTH functions.

A left parentheses FOLLOWED BY A SPACE is also a Forth word. It indicates that what follows the left parentheses is a comment. Comments are ended with the first occurrence of a right parentheses. In our coding examples above, comments are entered for your information and we do not intend for you to type in the comments, although if you are type-happy you could go right ahead and do so. Forth totally ignores comments and does not compile or use up any memory space for comments. Since Forth programs can get very cryptic, it is highly recommended that you sprinkle them liberally with comments at any point in the source statements. Remember that the left parentheses must be followed by a space.

When the Forth compiler encounters a sequence of characters delimited by a space, it first looks up the sequence in its "dictionary". Finding the sequence in its dictionary indicates that it is looking at a genuine FORTH WORD and it then acts according to the specification for that word.

If it fails to find the sequence then it tries to convert the sequence into a "number". In the first example above, 5629 is one "number". In the second example, 5 and 6 and 9 are three separate numbers. The 2 in the second example is actually a standard Forth word, a defined constant. The commonly used numbers 0, 1, 2, and 3 have been defined to the system as constants and serve to speed up compilation since the system does not have to search the entire dictionary and then attempt to convert the character string into a number.

Numbers in the range of -32768 thru 32767 are converted into 16-bit signed binary literals. Numbers from 32768 thru 65535 are converted into 16-bit unsigned binary literals. It is possible to deal with larger 32-bit binary numbers, but that will be a subject for a later discussion.

In the event that Forth cannot find a sequence of characters in its dictionary, and if it is unable to convert the sequence into a valid number, then it will go into abort mode and will reject the sequence with a question mark. Entering an invalid word, such as XXX, is sometimes useful when you want to reset the stack, and abort-mode will do just that.

Some versions of Forth are case-sensitive. To avoid making things any messier than they already are, keep your caps-lock key in the caps-only position when using Forth.

One final word on our tutorial coding scheme before we really get started. Since we will be covering both the ST and the 8-bit, which are using different versions of Forth, (versions that are 10 years apart), it will sometimes be necessary to use different instructions for the different systems. Any paragraph that begins with the characters ST. should be skipped by the 8-bit people. It will contain information that applies specifically to the ST or to the FORTH-83 version. Any paragraph that starts with 8-Bit. should be skipped by the ST users since it will contain information that applies to either the 8-bit machine or the fig-FORTH version of Forth.

Once we get beyond the initial startup information, the need to use the ST or 8-Bit paragraph designators will become very rare.

GETTING STARTED.

The 8-bit users will be able to get started a lot easier than the ST users, primarily because of the format of the club public domain distribution disks. So here we go with the first of the system specific paragraphs.

8-Bit. To bring up FORTH: insert club disk # 108 into your number 1 disk drive, disable Basic (by holding the option key on XL/XE models) and turn on the machine. The fig-FORTH msg should appear after the program loads. Note that Forth loads a lot faster than most DOS-based applications on bootable disks. Faster than DOS alone in most cases. After the program loads, type in the following:

WARN <cr>
DECIMAL <cr>

8-Bit. Please be sure to enter the DECIMAL command every time you load in the system, otherwise you may get very strange results when dealing with numbers.

ST. The ST users will first have to unARC the FORTH83.ARC file from ST disk # 126 onto a clean formatted disk. If you have single-sided drives, the complete set of unarced files will not fit on a single disk. Not to worry, simply delete the last file, (the incomplete one), and unarc that file separately onto another disk.

ST. If you are using a color monitor, set medium res, (80-col), mode and save the desktop on your new Forth disk.

ST. To bring up FORTH after unarcing: Place your new disk into the A drive and click on the program **F83KERN.TOS.** This is a bare bones version of Forth that will be best suited for our initial lessons on elementary functions. After the program loads, press the return key. Forth should respond with an ok message. Each time you bring up the system using F83KERN.TOS, it will be necessary to set **the ST screen to wrap-around mode**. Enter the following:

27 EMIT 118 EMIT <cr>

ALL. The **OK** message is Forth's equivalent of the BASIC READY message. OK means that Forth has completed the prior request and is now waiting for more user input from the terminal.

The next thing that we need to do is to make a backup disk to use as a working copy. The importance of this step cannot be overemphasized since it is very easy for a beginner to destroy a Forth disk. With Forth, you have the power to do anything your heart desires, and if you inadvertently tell Forth to overwrite something important, it will be happy to do so.

BACKUP

ST. This time the ST users have the advantage. To make a backup, simply copy your unarced disk to a blank disk using the desktop copy function.

8-Bit. Your Forth disk does not use standard DOS files. Thus it can NOT be copied by using DOS. You must use a sector copier or Forth itself to make a backup copy. The club now has an excellent PD sector copier on club disk # 110. If you have disk # 110, or another sector copier, then use it to copy both the front and the back sides of your Forth disk.

8-Bit. If you do not have a sector copier then initialize two blank disks by using the DOS single density format function. Then bring up the Forth system as described above. Type in the following:

BACKUP <cr>

8-Bit. Forth will then proceed to ask you to insert the source disk and then the destination disk. After about 5 disk swaps, Forth will announce that the copy is complete. **Do the same for the back side of your club Forth disk.**

ALL. If you fail to make a backup, then come around in a few weeks so that we can say, "I told you so!".

Lessons learned in this session:

1. How to bring up your Forth system.
2. How to make a backup copy of your Forth disk.
3. The difference between a number and a Forth word.
4. The definition of a Forth word.
5. The space delimiter.
6. The character set that can be used in a Forth word.
7. The definition of a comment.
8. How to reset the system.
9. A set of format codes that this tutorial will use.

LESSON # 1.

Now that we know all of the above, maybe we can start playing with our new toy. Bring up your Forth systems everybody.

Type in the following:

1 2 3 4 5 6 7 8 <cr>
(use a space between each digit)
.S <cr>
(note the period in front of the S)

Hopefully, after the first return key press, the system responded with an OK message, and after the second return it printed out the number sequence with another OK message. Somehow the system managed to remember the sequence of the numbers that we entered, and the .S word allowed us to print out that sequence.

Enter the following:

.S <cr> (again)

Looks like our number sequence is still there. When we entered the string of numbers, the system converted those 8 numbers into eight 16-bit binary values and stored them on something called the "**parameter stack**".

They are still there, unless you have done something that has reset the parameter stack, (remember ol' XXX ?).

XXX <cr>
.S <cr>

The parameter stack is the "heart" of Forth programming, so it is important to understand how it works. The stack functions like a holder for a stack of cafeteria trays. Placing an item on the top of the stack depresses the stack so that only the top item is visible. When we remove an item from the stack we retrieve the last item that was placed on the stack, and the removal will allow the next item to appear for the next retrieval. This is commonly known as a "**last-in, first-out**" (**LIFO**) retrieval sequence.

The parameter stack is the place where most Forth words will expect to find their input, and it is the place where most Forth words will leave their output after performing their function. Most Forth words will physically remove, **(eat)**, their input from the stack. This is in contrast to the action of the .S word.

The .S word displays the contents of the stack without disturbing the stack contents. It is normally inserted into a program for debugging purposes, and it may be the only debugging tool you will ever need.

Before we leave the subject of the stack, we need to say a few words about stack notation. Stack notation takes the form of a comment that every good Forth programmer should include in his source code. It is a form of shorthand that explains what a given word requires in the way of input, and what it will leave on the stack as output.

The stack notation for the + word has the following form:

(w1 w2 -- w3)

Everything to the left of the dash indicates the input that is required by the function. In this case, the + (plus) function requires two 16-bit signed binary values as input, w1 and w2.

Everything to the right of the dash indicates the output that the function will leave on the stack. In this case the plus function will leave a 16-bit signed binary value (w3) that is the sum of w1 and w2. Note that the plus function has eaten (removed) its input from the stack.

The stack notation has one more subtle indicator. The **right-most item to the left of the dash (w2)** is the item that should be on **the top of the stack at input time.** While this is not to important for the plus function, it can be very important for other Forth words such as "store".

The stack notation for ! (store) is

(x addr --).

This notation indicates that the memory address where we want to do the storing must be on the **top** of the stack when we use the ! function. As you may have surmised by now, x is the value that we want to store, and it must be one value beneath the top of the stack. Both x and addr are 16-bit binary values and **the only way the system can tell them apart is by their position on the stack.**

Now perhaps you can see the value of stack notation, and how you will be using stack notation to determine the order in which to place objects on the stack.

On the **output** side, the **right-most object will again be the top** of the stack. The stack notation for /MOD (divide-mod or division with remainder) is

(n1 n2 -- mod quot).

In this case the quotient output will be on the top of the stack and the remainder (modulus) will be beneath it. **n2 is the divisor.** Guess what n1 is.

We have been hitting you with some undefined n x and w characters. These are codes used to describe the type of value required. Rather than have you memorize the codes, and then forget them, let me refer you to the FORTH-83 reference chart that contains a Glossary of these codes and also a listing of the standard Forth words together with their stack notations. We will upload a file called FORTHREF.DOC that can be listed to obtain a copy of the chart. Last month's newsletter also contained a copy of the reference chart.

Don't let the number of words on the chart overwhelm you. Most of your Forth programming will be done with only a very small number of these words, and when you need to use an unfamiliar word, a quick reference to the chart and the associated stack notation will provide you with the information needed to use the unfamiliar word.

Play time again. Please enter the following:

XXX <cr> (clear stack)
DECIMAL <cr>
(make sure we are using base 10)
9 10 11 12 13 14 15 16 35 <cr>
.S <cr> (list stack in base 10)
HEX <cr> (set hexadecimal)
.S <cr> (list stack in base 16)
2 BASE ! <cr> (set binary)
.S <cr> (list stack in base 2)
DECIMAL <cr> (set base 10)
.S <cr> (list stack in decimal)
36 BASE ! <cr> (set base 36)
.S <cr> (list stack in base 36)
DECIMAL <cr>
(return to base 10 before we forget)

How about those apples. Most of you who have been around computers long enough to dare taking a look at Forth, have already heard of the base 10 (decimal), base 2 (binary) and base 16 (hexadecimal) numbering systems, so we will not bug you with another long-winded explanation of hexadecimal.

However, we bet you never heard of the base 36 numbering system before? This numbering system uses the characters 0-9, A-Z, to represent the decimal values 0-35. Don't ask me what it is good for.

Actually with Forth, you can set the number base to almost any value, although you will have difficulty using any base values above 36.

It is important to know which numbering system is currently in use. Each Forth word that changes the number base is duty bound to return the base to its previous state before terminating.

For some unknown reason the 8-Bit Forth is set up to start in HEX mode. **8-Bitters, don't forget to enter DECIMAL <cr>**

when you start out, otherwise anything you do with numbers will be in hex notation.

Items learned in this Lesson:
1. How the stack operates.
2. The .S word.
3. Stack Notation.
4. Identifying top item on stack.
5. Identifying input and output.
6. DECIMAL, HEX, BINARY and base-36 numbering systems.
7. In addition, we used several Forth words that we will cover again in detail in the later lessons.

Lesson # 2: Stack Manipulation

In the prior lesson, we learned about the operation of the parameter stack, the stack notation codes, and the importance of the position of items on the stack.

This lesson will cover the tools that allow us to reposition, or copy the stack values.

The most important tool at our disposal is common sense. We can look ahead and sequence the processing so that the next item we need is on the top of the stack.

This is the method you would normally use when coding a mathematical equation. Since the order of precedence in a Forth equation depends upon the order of appearance, (we do not use parentheses for precedence), then by processing the highest precedence first, (inner most parentheses algebraically), and by arranging the operands in the proper order, the next needed operand magically appears at the top of the stack.

The common sense method of stack ordering produces the fastest performing code and also the neatest looking and most trouble free code, mostly because you have spent some time thinking about the subject.

Part of the common sense method is to avoid putting to many items on the stack at the same time. A recommended level is five items or less. When you go beyond five items, then things begin to get confusing and errors creep in.

The rest of this lesson will be devoted to explaining the stack manipulation words. (See Forth-83 reference chart).

Be acutely aware of the difference between the words "duplicate" and "move" in the following descriptions. Duplicate means to make a copy of the stack item leaving the original item where it was, while "move" means to physically remove the item from where it was and place it in a new position on the stack.

DUP (x -- x x): DUPlicate is probably the most used word in the Forth language. Since Forth words will eat their input, we often need to save the input, to be used again, by duplicating the top of the stack.

DROP (x --): DROP will eat (discard) the top item on the stack. One of the debugging checks, that has served this author well, is to check the contents of the stack after execution. Quite often logic errors will leave extra unexpected words on the stack. Don't just willy-nilly DROP those extra words, identify them and correct the cause.

Keeping the stack neat and clean by using DROP where needed is simply good programming. There is a finite limit to the capacity of the stack and it can be overrun.

SWAP (x1 x2 -- x2 x1): SWAP the top two items on the stack. This should be self-explanatory. The second item on the stack is moved to the top of the stack.

OVER (x1 x2 -- x1 x2 x1): Like DUP only the second item on the stack is duplicated and placed on top. example:

OVER OVER (x1 x2 -- x1 x2 x1 x2) is useful when you have two items that are about to be eaten and you need to duplicate the original values for a subsequent use.

ROT (x1 x2 x3 -- x2 x3 x1): Pronounced rote, ROT reaches back to the third item on the stack and moves it to the top. ROT ROT ROT would perform the useless function of ROTating the top three items on the stack and leaving the stack as it was before we said ROT ROT ROT.

PICK (n -- x): n is zero-based with 0 being the top item on the stack. PICK duplicates the nth item on the stack and places it on top. 0 PICK is the same as DUP. 1 PICK is the same as OVER. If you find yourself using PICK a lot, you are probably placing to many items on the stack and you are also inviting maintenance problems since the depth of the stack is highly dependent on any changes to the existing code including any changes to underlying words.

ROLL (n -- x): Similar to PICK, except that the nth item is moved to the top of the stack rather than being duplicated. Again, n is zero-based.

?DUP (x -- x x) or (0 -- 0): If the top stack item is not zero then duplicate it. Pronounced "question-dup".

>R (x --) Pronounced "to-r". Move the top item on the stack to the return stack. Something we haven't told you about yet, Forth has more than one "stack". Besides the parameter stack, there is an additional stack called the return stack. The return stack is the place where Forth maintains its "return" addresses as it picks its way thru the program. The return stack is particularly sensitive during BEGIN and DO loop structures. >R moves the top item on the parameter stack and places it on the return stack giving you a temporary place to keep it with the emphasis on temporary. The item must be removed from the return stack before you do a loop-back, or if you are not in a loop, before you end the current word. Using >R is not recommended.

R> (-- x): Pronounced r-from. R> removes the top item from the return stack and places it on the parameter stack. This is the reverse of "to-r", and like to-r, its use is not recommended.

R@ (-- x): R-fetch; Copy the top of the return stack and place it on the parameter stack. Since the return stack is not being disturbed, this word is much less dangerous than >R or R>, still you will need to know a lot about the system before you can make anything useful out of this word.

DEPTH (-- +n) Count the number of items on the stack. Can be used to help bulletproof a program at the user interface level. Enables a word to check the stack for sufficient input before it attempts to use the input. Could also be used to condition a word to accept a variable number of inputs.

8-BIT. PICK and ROLL are not implemented on the 8-bit fig-Forth system, however, when we get into adding to the system, we will implement these words.

PLAYTIME. Please bring up your systems and enter the following. (ST users be sure you have set screen wrap-around.). Observe the effect the stack manipulation words have on the contents of the stack.

```
DECIMAL XXX <cr>
     ( decimal mode and clear stack )
22 36 DUP .S <cr>
     ( gives you 22 36 36 )
ROT ROT .S <cr>
     ( stacked item 36 22 36 )
OVER OVER .S <cr>
     ( 36 22 36 22 36 are you with us?)
SWAP OVER .S <cr>
     ( 36 22 36 36 22 36 stacked items)
DROP DUP OVER + SWAP
     ROT .S <cr>
     ( Dolly? Here comes the boss!)
DROP DROP DROP DROP
     DROP DROP .S <cr>
```

If you were able to follow what was happening on the next to the last entry, (the one that started with drop dup over), then you get a gold star. That line was purposely designed to make your head spin. For those of us who never get gold stars, try entering .S before and after each of the words on that line. This exercise should give you an appreciation of .S as a debugging command.

The drop-dup-over line also contains another lesson in Forth programming. One of the major pitfalls is to get carried away with stack manipulation, to the point where, we lose track of what we are doing. It would have been much simpler to just say 44 22 36, and the world, including myself, would have known just what was going on. K.I.S.S.

Items covered in this Lesson:
1. K.I.S.S. -- Keep it Simple Stupid, don't put more than 5 items on the stack.
2. Look ahead and plan the order of execution so that the stack is ordered properly when you need it.
3. Words covered: DUP, DROP. SWAP, OVER, ROT, PICK, ROLL and DEPTH.

To other Newsletter Editors:

As we receive more feedback from students, and find better ways of presenting the subject, then we will be incorporating those ideas and updating this document.

The latest text for this series of Forth tutorials is available on the DALACE BBS, (214) 255-8256, in file area #1, which is accessible to the general public. The referenced PD Forth compilers are also in the same file area. A nice thank you gesture would be to leave us a newsletter article in return. ED.

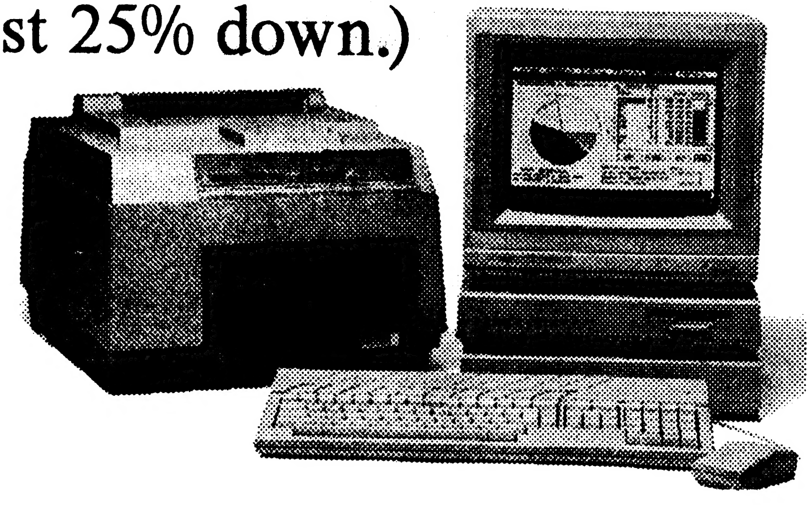
COMPUTER DISCOVERIES
ANNOUNCES
A SECOND LOCATION

Located at 1113 Pipeline, (at Melbourne), #125,
Ft Worth near Northeast Mall
Opening June 26th!!!

GRAND OPENING SPECIALS

MEGA2 ST CPU ONLY:	$1100 or $1045 Cash
MEGA2 ST W/MONO:	$1200 or $1140 Cash
1040ST MONO:	$ 699 or $ 665 Cash
MEGA FILE 30:	$ 599 or $ 570 Cash
MONOCHROME MONITOR:	$ 150 or $ 142 Cash

(NOTE: On items listed above which are not in stock receive extra 5% discount by ordering and paying at least 25% down.)



Most Wanted List

Dal-ACE Experts

Donny Arnold...289-6746...Call before 10 p.m. 8-bit General knowledge.

Eb Foerster...357-7602...Call from 7 to 10 p.m. Turbo Basic, SynFile, SynCalc, Assembly.

Ron King...(817)283-0674...Call from 5 to 10 p.m. 8-bit hardware.

John Saunders...(817)566-0318 C and Assembler languages.

Michael Trombley...429-6134 ST general knowledge.

Ralph Tenny...235-4035 Call from 7 to 10 p.m. ST general knowledge and hardware.

Rene Tucker 223-6176 8-bit general knowledge.

John Winer...907-1349 Systems Programming and general knowledge.

Infomart Directions

From North Dallas, take either Stemmons (I-35E) or the Dallas North Tollway south. From Stemmons, take the Oak Lawn exit, turn East, and park at the Infomart. If you are using the tollway, exit right on Wycliff, go left on Harry Hines Blvd. to Oak Lawn, and turn right. From the South, take Stemmons north, then follow above directions. Infomart is the big, white, steel and glass building south of the other marts. **GUESTS ARE WELCOME!!!**

Newsletter Advertisements

Personal ads are free to all current members. Please see the editor for details.

Commercial Rates

Full Page	$35
Half Page	$25
Quarter Page	$15
Business card	$10

For an additional $10 per full page, or $5 per partial page, you can request that your ad be placed on the inside front or back cover or the center page spread. This service is first come, first served.

Ads must be camera ready. Submission deadline for ad copy is the first of the month prior to publication date. That is, November 1st will be the deadline for your ad to appear in the December newsletter. Mail copy to the address on the back page, or contact the advertising manager, editor, or the Vice President of Communications. Copy received after the deadline will be run the following month. For contract advertisers, if no new ad is received prior to the deadline, the most recent ad will be run.

Editorial Policy

The editorial staff of the Dal-ACE newsletter reserves the right to edit your submissions for spelling, punctuation, grammar, clarity, and for reasons of space limitations.

Newsletter Submissions

Submissions are welcome in any form. Submissions can be uploaded to the club BBS or they may be submitted to the editor at the club meetings or by mail.

Disclaimer

The material printed in this newsletter reflects the opinions of the authors. Opposing opinions are solicited. Unless otherwise stated, the material in this newsletter is not copyrighted and no rights are reserved.

The purpose of this newsletter is to present information for your consideration. Neither the editor nor Dal-ACE make any claims for the validity or usefulness of this material. The reader is the final judge of any product or advice presented.

Infomart Meeting Dates

Firm Dates: June 10, July 8, August 12.

Tentative Dates: September 23, October 14, November 11, December 16.

Meeting Schedule

10:00-11:00...8-bit SIG
11:00-11:30...Disk sales
11:30-12:00...Main meeting
12:00-12:30...New Users SIG
....Newsletter Exchange SIG
12:30-2:00....ST SIG

DAL-ACE OFFICERS

President:	
Donny Arnold	289-6746
Vice President:	
Brenda Arnold	289-6746
VP Communications:	
Anita Uhl	492-8682
Secretary:	
Michael Duke	739-3116
Treasurer:	
Rene Tucker	223-6176
Members at Large:	
Terry Borchardt	296-4699
Dave Gramm	(214)370-7143
Randy Randolph	381-7624
Marc Salas	717-4615
~~Ralph Salmeron~~	~~254-8633~~
Jim Jackson	

DAL-ACE BULLETIN BOARD

(214) 255-8256

DAL-ACE VOLUNTEERS

Editor:	
Open	
Ad Manager:	
Marc Salas	717-4615
BBS Sysop:	
Jim Jackson	864-0805
8-Bit Librarian:	
Tim Mixon	356-4725
ST Librarian:	
Gary Loges	(817)481-4186

ALL MATERIAL PRINTED IN THIS NEWSLETTER MAY BE REPRINTED IN ANY FORM PROVIDED THAT DAL-ACE AND THE AUTHOR, IF APPLICABLE, ARE GIVEN THE PROPER CREDIT. LIKEWISE, PORTIONS OF THIS NEWSLETTER MAY BE REPRINTED FROM OTHER NEWSLETTERS AND ARE SO NOTED.

DAL-ACE INC.

Dallas Atari Computer Enthusiasts

Dal-ACE is an independent user education group that is not affiliated with the Atari Corporation. This is the official newsletter of Dal-ACE and is intended for the education of its membership as well as for the dissemination of information about Atari Computer Products.

Dal-ACE membership is $20 per year. BBS only membership is $10 per year. This newsletter is written, edited and published by volunteers. Its availability and/or distribution may, at times, be subject to circumstances beyond the control of the club officers. A pink address label indicates that your membership expires this month. Other user groups may obtain a copy of this newsletter on an exchange basis.

DAL-ACE

P.O. BOX 851872
RICHARDSON, TEXAS 75085-1872