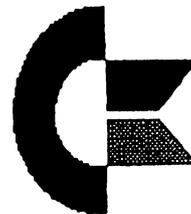
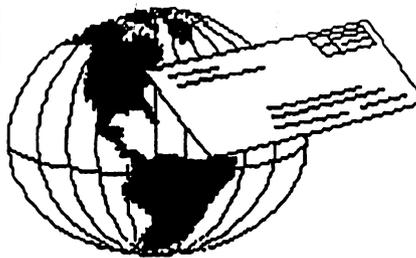


# COMMODORE MAILINK

**NEWSLETTER OF —  
MEETING 64/128 USERS  
THROUGH THE MAIL**



## **MEETING NEWS MARCH 2005**

Our 2005 membership stands at 91 as of late February. We almost managed to maintain our 100-some membership that held relatively steady for some years, in spite of declining Commodore user group memberships elsewhere. While our club will never be all things to all people, we are interested in hearing your views. What do members want to read about and learn in the pages of your Commodore Mailink newsletter? Machine Language? WINGS? WHEELS? geoProgramming? Using a PC as a slave to your Commodore? Fun Graphics Machine? Jeri Ellsworth's ongoing C-1 development? Getting online with a stock Commodore? Getting online with a souped-up, maxed-out Commodore? Using a Commodore to plot the trajectory of NASA's next space launch? Using a Commodore to solve the ancient riddle of the Pythagorean Table? Whatever your Commodore interests, nobody in this group will know unless you tell us. Remember the contents of our newsletters are what you, the members, dictate, edit, and create.

We all welcome new member, Clay Howard, whose application reached us in time to be included in our March BIO's. This is a good time to revisit the BIO's. Our BIO's are short biographical sketches of members which help us connect with others of similar interests. If you look closely in the left margin of the BIO's, you will notice those small, solid, black boxes adjacent to occasional names. Those members have volunteered to be "Friendly Correspondents" in the event some other member has an inquiry or comment. This is especially helpful to new members. But remember, those names not so denoted are not necessarily unfriendly! They have just not elected to be specifically so listed.

A new address has been received for Joan Cancelli which is 5 East Avenue, Spencer, MA 01562-2939.

Hope you enjoy this issue which contains our regular columns on TWS, geoPublish, BASIC, Questions and Answers, plus an article on Machine Language programming which is a welcome addition, and more of the same would be welcomed. OK, enjoy the read.

Linda Tanner, President

**BUSINESS OFFICERS:** (addresses in BIO's)

**PRESIDENT :** Linda Tanner, tannerlj@yahoo.com

Handles group business, compliments, complaints, threats, etc.;

**VICE-PRESIDENT:** Francis Redmond, redinc@earthlink.net;  
sends "late" reminders;

**TREASURER:** Emil Volcheck, Jr. emilv@mercury.ccil.org  
receives dues, donations, balances bank account, disburses monies;

**MAILINK-ON-DISK EDITOR & MAILER:** Richard Savoy;  
RSavoy5578@aol.com; compiles and archives Mailink On Disk and  
orchestrates photocopying and mailing of CML;

**ARCHIVIST and MANAGING EDITOR:** David Mohr,  
lordronin@videocam.net.au; oversees receiving, maintaining and  
copying CML archives; backup editor;

**MEMBER BIO EDITOR:** Brian Vaughan:  
frogissam@neteze.com; edits member addresses and BIO's; denotes  
member as "Friendly Correspondent" in BIO's at member request;

**E-ADDRESS EDITOR:** Joseph Fenn, jfenn@lava.net;  
maintains email addresses of members; posts list online;

**WEBMASTER:** Andrew Schwartz, andyschwartz@hotmail.com;  
maintains Mailink website, <http://mailink.videocam.net.au>;

**QUESTION & ANSWER MAN:** Rolf Miller; answers C= related  
questions of wide variety; rolfmiller@aol.com;

**YELLOW PAGES EDITOR:** Jorge Pedreira,  
edar43@abaconet.com.ar; maintains resource lists of C= oriented  
products and services; produces March/September Yellow Pages.

#### **THE EASY WAY TO CONTACT OFFICERS:**

president@MAILINK.videocam.net.au

vicepresident@MAILINK.videocam.net.au

treasurer@MAILINK.videocam.net.au

bioed@MAILINK.videocam.net.au

emailed@MAILINK.videocam.net.au

webmaster@MAILINK.videocam.net.au

managingeditor@MAILINK.videocam.net.au

diskeditor@MAILINK.videocam.net.au

editor@MAILINK.videocam.net.au

#### **COMMODORE MAILINK POLICIES and RULES:**

The Commodore Mailink (CML) is published every other month by Meeting 64/128 Users Through the Mail, copyright 2005 by Meeting 64/128 Users Through the Mail. All rights reserved. Permission given to reprint if credit is given to the Meeting 64/128 Users Through the Mail. The names, "Meeting 64/128 Users Through the Mail" and "Commodore Mailink" and "CML" are also copyrighted. Any and all opinions expressed in this publication are the views of the authors, and in no way necessarily reflect the viewpoints, attitudes or policies of Meeting 64/128 Users Through the Mail, unless so stated or indicated. Neither Commodore Mailink nor Meeting 64/128 Users Through the Mail condones piracy of copyrighted software or other material. All programs published are with the permission of the author, or are to the best of our knowledge, in the public domain. Software offered for sale is said by the seller to be either public domain, or if commercial, is the original disk with original documentation. All manuscripts or any material for review or publication should be sent to the editor of the next issue. Commodore Mailink reserves the right to edit submissions.

Members may place free Commodore BUY/SELL/TRADE ads in the Mailink. Send diskfile of ad to next editor, or if ad is very short, hard copy may be submitted instead

#### **EDITOR GUIDELINES:** (abbreviated version))

Editing a newsletter involves collecting articles, programs, and other items, then placing these items in printed newsletter form. An editor may opt for the "cut and paste" approach, where articles, once printed, are literally cut and pasted onto each "master page". Or, publishing software may be used.

Editor should where possible use the two-column per page format for better readability. Also, all page margins, excepting page 2, should be no less than 3/4 " on each right and left margins. Text, regardless of which font, should, with rare exception, be 12 point or larger. The number of different fonts on a page is not written in stone, but no more than two is a good rule of thumb, or three if you include footers and/or headers.

Requirements for the Editor include, regular columns such as TWS, geoPublish Tutorial , BASIC, and Q&A DESK, and if submitted, ARCHIVES, BUY/SELL/TRADE ads, Treasurer's Report, Obituaries, Announcements, Address Changes, and New MemberBIO's. Also the Editor must detail the computer system, including software and hardware and printer used in the production of the current newsletter, as well as name, requirements, and deadlines of next editor. This could be in two separate columns: "The Editor's Desk" and "The Next Editor". If as Editor, you receive more than enough to fill pages 3-18, you will publish all "requirements" and some of the "options", and forward to next editor only whatever "options" would not fit in newsletter.

The Editor should edit each item as it arrives, creating a diskfile and a backup diskfile, making certain a TWS version of all disk files is (also) created. A good, clean "master copy " on single, unfolded sheets, printed on one side only and protected by cardboard should be sent along with the TWS disk files to our Mailer, Richard Savoy, by the first day of the month you are editing. **NOTE:** to include a color page, send 100 completed sheets of that page, printed on both sides (ready for insertion into CML), along with the "masters" and TWS disk. A second copy of CML (copied on the back of junk mail is OK) should also be sent to the President for proofreading purposes.

Always keep your backup disk in a secure place until the Mailink is in the hands of members. If you suddenly realize it is near the first of the month, and you are not nearing completion of the newsletter, notify the President. It is much easier to deal with a problem if it is made known.

Editing can be fun. It can be mentioned in your resume', so what are you waiting for? Contact us now.

**NOTE:** pages 1, 2, 5 and 6 of this newsletter were created with a C128, SCPU, CMD HD, RamLink, Lexmark Optra 40 PS II printer, geoWrite, geoPublish, Wheels, and PostPrint 3, by Linda Tanner.

March Editor: **Richard Savoy**

# Jim Collette GEOS software now Public Domain

Bruce Thomas wrote on January 17, 2005:

"I think it might be beneficial to MaiLink readers if this notice was printed."

January 16, 2005

Jim Collette and Maurice Randall have agreed that GEOS Software written by Jim, including the collection distributed as the Collette Utilities Disk by Click Here Software (<http://www.cmdrkey.com>), is no longer viable as a commercial product and can now be considered Public Domain.

The Collette Utilities Disk is available in .D64 format on the download page of the CUE web site (<http://www.edmc.net/cue>) and will be added to the next version of the geoSpecific PD/Shareware CD.

The Collette Utilities Disk contains some of Jim's best GEOS software but a number of the utilities have been superseded by functionality in the Wheels upgrade to GEOS and by the PostPrint program. Both Wheels and PostPrint have been written by Maurice Randall.

PostPrint program. Both Wheels and PostPrint have been written by Maurice Randall. Programs included on the Collette Utilities Disk include:

Font Editor 2.5 - create standard GEOS fonts, MEGA fonts and Perfect Print fonts

Font Changer - change fonts and point sizes in geoWrite documents

geoWizard - run one GEOS program without closing one already in use

geoWizDump - screen dumper for use with geoWizard

MiniDesk - Desk Accessory (DA) for copying and deleting files

Select Printer - DA for changing Printer Drivers

DOS\_Wedge - DA for sending commands to disk drives

AutoAlbumAdd - DA for easier handling of Photo

DOS\_Wedge - DA for sending commands to disk drives

AutoAlbumAdd - DA for easier handling of Photo Scraps

PS.Processor - create customized PostScript output

PS.Patch (disk) - patch geoPubLaser for disk output

PS.Patch (GC) - patch geoPubLaser for geoCable output

LaserWrtr 2.1GC - geoCable version of Laser-Writer Printer Driver

This collection of programs will greatly enhance your use of GEOS. Some of the programs, not including geoWizard, work as written under Wheels. If you want to use geoWizard with GEOS make sure you read the documentation first.

Make sure you read the 'Readme PD' file on the disk for information on Easter Eggs in geoWizard.

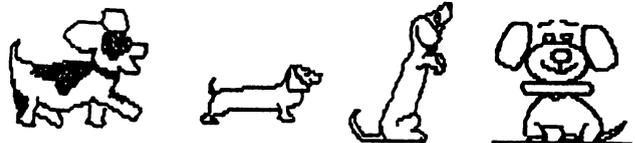
## NEXT EDITOR FOR MAY

By Andrew Schwartz

Articles can be sent on either 5.25" or 3.5" floppy disks, or via the Internet, if mailing hard copies, be sure they aren't folded. If you have any questions contact at my Snail mail or Internet address listed below. Deadline is April 15 for mailing.

Andrew Schwartz  
6300 King Louis Drive  
Alexandria VA 22312

Email: [andyschwartz@hotmail.com](mailto:andyschwartz@hotmail.com)



# COMMODORE VEGAS EXPO RAFFLE PRIZES

BY: BRUCE THOMAS

Robert Bernardo has previously passed along info on our planned Commodore Vegas Expo (CommVEx) to be held in Las Vegas, Nevada on July 30 and 31, 2005. Check the CommVEx web site (<http://members.shaw.ca/CommVEx05/>) for all of the details now and updates as summer approaches.

Robert Bernardo has also pointed out that 2005 is the 20th Anniversary of the Commodore 128.

I would like to pass on some info about some of the raffle prizes we have rounded up so far. We are continuing to try to add more raffle prizes and will announce them as we round them up.

- geoSpecific GEOS PD/Shareware CD compiled by Bruce Thomas (see <http://members.shaw.ca/cue64>)
- Compute!'s Gazette CD - scans, .d64 images, etc compiled by Jeremy Sieracki.
- ReRUN Compendium CD - created by Michael Hunter. Marketed by VideoCam Services under licence from Maurice Randall (see the Central Market at VideoCam <http://cbm.videocam.net.au/>)
- Commodore 64 DTV

If you win one of the subsidiary prizes your raffle ticket will go back in the draw drum before the final raffle prize draw so you have a chance at it as well.

**So what is the final raffle prize at CommVEx you ask?**

Some of you may recall in December when Michael Hunter, of 8BitDesigns fame, announced that he was offering for sale a customized, gold painted, C-128. I can't describe all of the features of this system here (see the CommVEx web site for all of the details) but it has 64 KB VDC RAM, a dedicated mouse port, stereo SID chips,

built in geoCable compatible Parallel Printer interface plus one of Michael's Power Pro 250 customized (110 Volt) power supplies with leads to power the C-128 plus 10 accessories (like a RAMLink and disk drives).

In a message to Homestead, Alan Reed mentioned that he thought this custom 128 should travel to EXPOs to be seen and admired by lots of people. This got me to thinking and I bought the Gold C-128 from Michael. But I don't want it.

OK, Bruce. Are you trying to say that this unique, highly customized, gold painted C-128 system is the main raffle prize at CommVEx? No. It is only PART of the prize.

Michael has also painted up two (yes, 2!) 1571 drives in gold to match the C-128. Both of these drives are JiffyDOS equipped. Both have been modified to get power from the Power Pro 250.

Michael has also painted a 1351 Mouse.

Also part of the package are replacements for all the chips inside the C-128. They have all been socketed so if the need should arise it will be easy to pull the chip and replace it from the spare pool.

A set of manuals will accompany the computer.

Either a 1084 or a 2002 monitor (won't be painted gold) will also be included in the package along with a set of Altec Lansing speakers to enjoy the Stereo SID output.

We are trying to make this system as complete as possible and a complete list will be available, with pictures, on the CommVEx web site once I receive everything from Michael.

Continue on page 7



# BASICALLY SPEAKING

by Linda Tanner

This time we are looking at "standard hi-resolution" bitmapping of the 40 column screen, using C128's BASIC 7.0 and the 64's BASIC 2.0. BASIC 7.0 has many more commands which make programming easier, but it can "spoil" a potential programmer who has never programmed in BASIC 2.0. For example to draw a circle on the screen using 7.0, one merely issues the "CIRCLE" command with the appropriate parameters. To draw an on-screen circle using BASIC 2.0, one needs to do that old thinking cap once again, and calculate how and where to place each point in the circle, i.e., how and where to turn on the appropriate bits that will make a circle appear on the screen.

But before we think of bitmapping circles or any other graphics on the screen, we have some "housekeeping" chores that must always be performed. I have used the same variables as seen in our "Guide" (below) so beginners will more easily understand. `BASE=2*4096:POKE53272,PEEK(53272)OR8` starts our bitmap at 8192, and note that only bit 3 is significant in bitmap mode, thus we use "OR8". Then to actually turn the bitmap "on", we need to turn on bit 5 of register 53265 via: `POKE53265,PEEK(53265)OR32`. Next, we must clear our bitmap so as to start with a clean slate. We do this by POKE-ing a zero into each of our 8000 bytes with this: `FORI=BASETOBASE+7999:POKEI,0:NEXT` and we then select a screen color, such as green with: `FORI=1024TO2023:POKEI,5:NEXT`. Now for the fun part.

Over the years I periodically tried the semi-circle program on page 127 of Commodore 64 Programmer's Reference Guide, and since it did not work, I assumed I had made a typo. Finally this year I looked at the code and there is an error. It merely needs one more statement to draw a semi-circle: `116 NEXTX` will complete the program. The program can be tweaked to produce a circle but we'll use a different methodology.

In working to create an on-screen bit-mapped circle, it did not take long for me to realize what other programmers discovered long ago. That is, when bitmapping the Commodore's 40 column screen, one must take into account this asymmetric 320 X 200 bit screen (see page 125 in the above Guide). My first "circle" looked more like an ellipse because a given number of pixels spread horizontally on the screen will appear to be of a different length when spread vertically on that same asymmetric screen. To demonstrate this phenomenon I've included a two-block routine as the first program, "LINE64-1.0" in which the horizontal and vertical lines supposedly contain the same number of pixels (they do), yet the lines are obviously of different lengths, which we can verify with a ruler. We must deal with this asymmetry in order to create circles that actually look like circles.

Once we have written the code necessary to create a circle, we still must be able to turn on only particular bits. I used the exact formulae on pages 125-126 in the GUIDE for turning on the specific bits desired, although I did condense them. For example, once your program determines "x" and "y" values, it then turns on the desired bits with:

```
30 BY=BASE+320*(INT(Y/8))+8*(INT(X/8))+(YAND7)
32 POKEBY,PEEK(BY)OR(2^(7-(XAND7)))
```

If you study pages 125 and 126 carefully you will see what is being done. If you do not understand it (I didn't the first time I read it either) break it down into parts and study only one portion at a time. (Rome wasn't built in a day.)

Our second program, "CIRCLE64-1.0", takes advantage of two basic trig functions, sine (sin) and cosine (cos) in creating a circle. Simply, we use a right triangle, and label the vertical leg of that triangle "y", and the hypotenuse, "r". The value of "y/r" for a given angle is a constant number mathematicians call the sin. Likewise, if we label the horizontal leg of the triangle "x", and we know that the cos of a given angle is merely the value of "x/r", then we can solve problems, including creating circles on computer screens. Here we are allowing "r" to be the radius of our circle. Since our radius is constant, and since our program causes minute changes in the angle (AN) value as it proceeds through the FOR/NEXT loops, with each new AN(gle) value, the computer calculates a new "y" and a new "x". The computer also handles these trig functions when "called", as in: `Y=100 + R*SIN(AN)`. The sketch on page six briefly demonstrates how triangles and trigonometry can help create circles.

Our third program, "CIRCLE128-1.01", is for the C128 and because of BASIC 7.0's additional keywords, is almost "idiot friendly". This isn't always bad, but remember the more you tax those neurons, the more efficiently they will function. In this program, we merely set X to 100 and Y to 40, set colors, then call the CIRCLE function--all in one line.

Our fourth program, "CIRCLE128-1.0" creates a larger circle that moves partway across the screen, disappears, then is followed by a smaller circle which also moves diagonally on the screen.

BASICALLY SPEAKING, continued from page five:

"LINE64-1.0"

```
10 BASE=2*4096:POKE53272,PEEK(53272)OR8:POKE53265,PEEK(53265)OR32:GOSUB44
12 FORI=1024TO2023:POKEI,14:NEXT:Y=100
14 FORX=110TO210:GOSUB30:NEXT
16 X=160:FORY=50TO150:GOSUB30:NEXT
17 GETQS:IFQS=""THEN17
18 POKE53265,PEEK(53265)AND223:STOP
30 BY=BASE+320*(INT(Y/8))+8*(INT(X/8))+(YAND7)
32 POKEBY,PEEK(BY)OR(2^(7-(XAND7))):RETURN
44 FORI=BASETOBASE+7999:POKEI,.:NEXT:RETURN
50 REM-LINE64-1.0-LINDA TANNER-CML-MARCH05--HIT RUN/STOP RESTORE TO EXIT
```

"CIRCLE64-1.0"

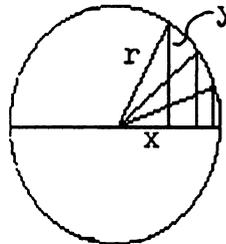
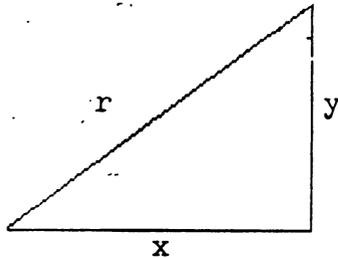
```
10 BASE=2*4096:POKE53272,PEEK(53272)OR8:POKE53265,PEEK(53265)OR32:GOSUB44
12 FORI=1024TO2023:POKEI,14:NEXT:R=70
14 FORAN=1TO90STEP.25:Y=100+R*SIN(AN):X=160+1.22*R*COS(AN)
16 BY=BASE+320*(INT(Y/8))+8*(INT(X/8))+(YAND7)
18 POKEBY,PEEK(BY)OR(2^(7-(XAND7))):NEXT
20 GETQS:IFQS=""THEN20
22 POKE53265,PEEK(53265)AND223:STOP
44 FORI=BASETOBASE+7999:POKEI,.:NEXT:RETURN
50 REM-CIRCLE64-1.0 CML3-05-L.TANNER--HIT RUN/STOP RESTORE TO EXIT
```

"CIRCLE128-1.01"

```
10 X=100:Y=40:GRAPHIC1,1:COLOR4,4:COLOR0,4:COLOR1,1:CIRCLE1,X,Y,Y,Y:REM--40COL--
CIRCLE128-1.01- MARCH05CML-LINDA TANNER-- RUN/STOP RESTORE TO EXIT
```

"CIRCLE128-1.0"

```
10 X=100:Y=40:GRAPHIC1,1:COLOR4,4:COLOR.,4:COLOR1,1
12 FORA=1TO2STEP.1:CIRCLE1,X*A,Y*A,40,40:GRAPHIC1,1:NEXT:GRAPHIC1,1
14 FORA=1TO900STEP10:SCALE1:CIRCLE1,X+A,Y+A,40,40:GRAPHIC1,1:NEXT:GRAPHIC1,1
16 SCALE1,1000,5000:CIRCLE1,180,100,100,100:GRAPHIC1,1:FORA=100TO200:CIRCLE1,180,A,100,100:
NEXT:GRAPHIC1,1
18 SCALE1,5000,1000:CIRCLE1,180,100,100,100:GRAPHIC1,1:FORA=100TO200:CIRCLE1,A,100,100,100:
NEXT
20 REM--40 COL--CIRCLE128-1.0-MARCH05-CML--L.TANNER-HIT RUN/STOP RESTORE TO EXIT
```



NOTE re "CIRCLE64-1.0": The "r" hypotenuse of triangle on left is the same "r" radius as seen in each of the three triangles overlaying the circle on right. With each increment of our angle (AN), in our imaginary triangle that is superimposed upon our circle, a new "x" and a new "y" will be calculated from our knowns: radius, AN, sin and cos. In line 14 the "1.22" is an "eccentricity factor" that compensates for C64's screen asymmetry. Not drawn to scale.

We are trying to make this system as complete as possible and a complete list will be available, with pictures, on the CommVEx web site once I receive everything from Michael.

Raffle tickets are selling now for US\$10.00 each or 3 for US\$25.00 if paying by cheque or (International) Money Order in US funds. Send your Cheque or Money Order to:

Robert Bernardo  
502 Whitney Lane  
Visalia, CA, U.S.A., 93277-1940

If you want to order tickets and prefer to use PayPal you can do that too but the prices will be US\$10.75 for one ticket or 3 tickets for US\$26.50 to cover PayPal overhead. Send your PayPal orders to rbthomas59(at)shaw.ca.

Raffle tickets orders before CommVEx must be received by July 15, 2005. Send your orders in early. We are not responsible for orders lost or delayed in the mail. PayPal is the preferred method of payment for pre-ordered tickets.

Along with your payment send your mailing address and e-mail address if you have one. A phone number might be nice to include as well.

Ticket prices at CommVEx will be the same - 1 for US\$10.00 or 3 for US\$25.00.

If you are in attendance at CommVEx and your name is drawn on Sunday you can take the system home with you. If you aren't in attendance we will ship the system to you anywhere in North America. If you live outside North America you will have to cover additional shipping costs to get the units to you. The system will be in 3 large boxes.

Profits from raffle ticket sales will go to fund future EXPOs as well as other Commodore related activities that many people benefit from.

Continued next column

Thanks for your help in supporting the Commodore Community and Commodore related activities.

**Bruce Thomas**

## QUESTION & ANSWER DESK

Rolf Miller, Q & A editor  
rolfmiller@aol.com

Questions may be submitted by regular or e-mail. Answers will be returned as soon as obtained, then published in coming issues, though the identity of those asking questions will not appear because duplicate, similar, and associated items will be compiled.

Q. My 1902 monitor has gone so far out of focus I can no longer read the text. No adjustment of the front controls resolves the problem, and I can see no focus control on the front or rear. Before I take it apart or pitch it, are you aware of any internal control for focusing this monitor?

A. According to several online sources, most monitors have potentiometers for adjusting brightness and focus integrated somewhere on the circuit board. However, degrading focus over time tells that things are getting old, and making adjustments may not help, indicating a spent picture tube or bad flyback, or both. If a new picture tube is needed, replacing the flyback will not produce acceptable results, and that suggests replacing the monitor.

(Users of 128's are reminded that an 80-column screen, though black & white, can be displayed on any monitor which allows for composite input, such as the 1702. This requires a cable with pin 7 of the 9-pin computer-side D-connector attached to the center pin of an RCA plug, and pin 2, which is ground, attached to the shield.)

Q. I have a type-in text-adventure program and would like to have images, like a Koala or Blazing Paddles image, show up for a few seconds, such as when a monster is met, then return to the program (I've seen programs that do this).

Continued on page 8

Continued Q & A from page 7:

I don't know how to make the image show up and return to the same spot in the program. Ideas?

A. A common way to make a BASIC program go do some repetitive thing then return is to utilize the GOSUB and RETURN statements. GOSUB causes program execution to jump to a subroutine that does the thing desired, then RETURN at the end of the subroutine returns execution to the main program. The format is as follows.

```
100 main
110 program
200 GOSUB 1000
210 continue
300 main
400 program
500 GOSUB 1000
510 continue
600 main
700 program
900 END
1000 subroutine
1900 RETURN
```

Of course, to accomplish any high resolution display in seconds, the subroutine needs machine language. The SYS statement is the usual means of jumping from BASIC to a machine language subroutine. Assuming the machine code resides in memory beginning at location 49152, SYS 49152 would be used instead of GOSUB.

Now, rather than writing the machine code from scratch, it is likely that a display routine from another program can be appropriated. Just make sure it ends with decimal code 96 (op code RTS), which is the machine language equivalent of RETURN.

Q. The January 2005 Mailink spoke of the 64DTV (a joystick with a built-in "64 on a chip" containing some 30 Commodore 64 games). But it showed no way for me to buy it. Is there a street address of the company selling this product?

A. As mentioned in the issue, the 64DTV available from QVC, which is a cable TV merchandiser. Their order phone number is 800-345-1515 (their web site also lists 888-345-5788). The single unit item number is E22494 for \$33 plus \$5.22 S&H. The item number for two is E25376 for \$57 plus \$5.22 S&H. Before ordering, however, you might want to check the toy stores, as some have reported finding them for less.

The street address for QVC is: 1200 Wilson Drive at Studio Park, West Chester PA 19380. Whether they accept mail orders is not known.

### EVERYBODY GETS ONE!

By: Richard Savoy

Kenneth Barsky spends a lot of time compiling, creating, editing groups of Commodore Computer material for others and himself. It goes back as far as I've known him from the early 1990's. He would make monthly music calendars for the Bronx Users Group monthly disk and I'd trade my CRCCC, monthly disk.

Once in awhile Ken will ask if I'd like some software that he has developed (and I never refuse) and try to put it in the CML disk version which is a 1541 double side disk with the CML newsletter in sequential file on the front side of the disk and Ken's software with other public domain programs go on the back side of the disk.

He recently offer me program, "Word Power" with about five years of quiz material plus other word games that took 1323 blocks with it own Menu program on a 3.5" disk. Great except one problem several of our members do not have 3.5" disk drives. So I put it on a 1571 drive which loads both side from the front side (no turning the disk over)! Like "The Oprah Show" Every current member gets one! One catch. CML disk version buyer will receive with the March disk issue. All others need to send me a Sixty Cent US postage stamp. **Richard Savoy**  
**Address in the BIO's list**

## BUY - SELL - TRADE

### CINCINNATI COMMODORE COMPUTER CLUB

Official Users Group Number 292  
c/o 31 Potowatomie Trail  
Milford, OH 45150  
[www.geocities.com/c64-128-Amiga](http://www.geocities.com/c64-128-Amiga)  
[cbmusers@yahoo.com](mailto:cbmusers@yahoo.com)

#### CLUB AD

The Cincinnati Commodore Computer Club has 1000's of used commercial items for sale at very reasonable prices. The lists include software, hardware, accessories, books, magazines and manuals. Because of the low prices we ask that buyers pay postage. For a 5 1/4" disk of the lists, send a floppy mailer to Roger Hoyer, 31 Potowatomie Trail, Milford, OH 45150. If you'd rather receive the lists by Email, contact Roger at [thunderbird@iglou.com](mailto:thunderbird@iglou.com) or the club at [cbmusers@yahoo.com](mailto:cbmusers@yahoo.com). A third alternative is to view them and download them from our web site- [www.geocities.com/c64-128-amiga](http://www.geocities.com/c64-128-amiga).

The club also sells the following printer tractor feed items:  
3.5" x 15/16" address labels in pastel shades of blue, green, pink and yellow, plus white - \$1.00/100.  
2.75" x 1 15/16" labels for 3.5" disks - \$1.50/100.  
3.5" x 6" postcards - \$1.50/100.  
Prices include postage.

**COMMODORE  
YELLOW PAGES  
WITH THIS ISSUE!  
YOU CAN'T MISS  
THEM, THEY ARE ON  
YELLOW PAPER  
TO REMIND YOU TO  
KEEP THESE IMPOR-  
TANT RESOURCES.**

**ALSO, THE MUTTM  
MEMBERSHIP BIOS  
ARE IN THIS ISSUE  
WITH ADDRESSES  
FOR EACH MEMBER  
& IMPORTRANT TO  
KEEP HANDY. BOTH  
ITEMS ISSUE ONLY  
TWICE A YEAR.**

## RESPONSES TO NOTE

By: Joe Fenn & Emil Volcheck

Prepared By: Richard Savoy

The following is from email between Joe Fenn and Emil Volcheck last November when Joe was renewing his membership in the group, I was one of the officers that received copies of the email-R.Savoy, ed this issue.

"Hi Emil,

Just a short note. Keep the excess as a donation. I find reading the bulletin nowadays somewhat of a mystery. Also on comp.sys.cbm. Many things like d64, x15410 cables, CDTV etc. Are all mysteries to me: I have 3 laptops, 2 NECs and 1 Tosh CD420, that I have divided my computer time in learning dos6.22 dos7 etc.; so I know the underpinings and often use dos on the PCs to get things done faster." Joe Fenn.

Joe, I think your point is well taken, you are probably not alone amongst MUTTM members regarding some of the developments in C= or associated stuff that many take for granted. I've noted your comments to Lord Ronin, (David Mohr) who is the editor of the next CML, mayhap he can start a trend to explain things that may be a bit arcane (maybe a glossary???)

"However in the process I have not kept up to date on CBM stuff as much even though the C128 is still the work machine for all that I do. I use packet radio with one, and internet on another. I tried recently to use a never used disk Novaterm 9.6 I bought and never even checked it out. DIALOGUE is my term used for years now. Novaterm has a setup file but it lists such things as HART when defining a modem. I use Aptrotek C24. I set it to HART whatever that means but in reading the book it said HART is only for speeds above 2400 baud, like swiftlink cartridge useage etc."

Novaterm 9.6 is a 64-only based program (with the ability to use the 80-column display when used on a 128 in 64-mode). I only used it to answer questions from 64 users at our local club meetings. I don't see any real merit compared to Dialogue 128.

The HART is an RS-232 interface like Swiftlink and Turbo (the CMD version of Swiftlink). I never heard of anyone who had or used it - but-somebodies must have <g>.

"To make things worse Win XP has no dos that you can get to, so I wasted about 3 years on learning DOS apparently. I only use the laptops for about 3 years on learning DOS apparently. I only use the laptops for playing Solitaire while making backups of the Ramlink (grin).

**Editors Remarks:** This was not the end of the message, but the part that was clearly about Commodore. David Mohr (Lord Ronin) did write in the January issue a article titled, "D64 and other Terms" starting on page 9 of that issue. RJS—ED

## EQUIPMENT CARE

**Joysticks:** Those joysticks that won't fire or don't in a particular direction can be fixed by opening the case and checking the five contact arms inside. Cleaning and slightly bending the spring-metal back up can quickly put the joystick back into action and you can get back to the task.

**Printer paper feed:** Keep the right hand tractor on your printer loose. This will reduce occurrence of the annoying problem of paper crumpling as it tries to climb up over the sprockets.

**Validate Regularly:** If you are scratching and saving programs regularly on disk ( or using the save @ - with replace) then make sure that you validate the disk regularly.

# GEOPUBLISH TUTORIAL PART 9 COLUMN ANALYSIS

By: Bruce Thomas

## Analysis

So far we have used a variety of column layouts. Have you noticed the differences? Have you seen one that is better than the others? Have you seen one that is worse than the others? Let's do a little comparison and analysis and see if we can identify an ideal layout.

We laid out Page 1 in a 2 column format using left justification and 5 point gutters. If your printout is like mine there are a couple of spots where the left column text almost runs into the right column text. This often crops up when printing with these settings and is one reason why I prefer using 10 point gutters.

For Page 2 we changed the layout to use full justification. This format can provide a cleaner look to the page and almost always assures us that the text will not 'run over' to the next column as on Page 1. There are, however, 2 problems here. GeoPublish does not do auto-hyphenation so this can create a lot more work for you, the user. The second, and maybe more important, issue is that full justification works by inserting, or removing, extra spaces between the words to stretch, or compact, them to fill the entire line width. This method makes it harder for the human eye to read the text as we have to work to read past the uneven spacing. I'm not saying that you are wrong if you use full justification but there are drawbacks.

On Page 3 we changed to 3 columns, fully justified with 10 point gutters. The shorter lines are easy to read without a lot of eye movement but the full justification makes the need for hyphenation even greater. My feeling is that full justification should only be used on 2 column layouts, if at all.



Page 4 kept the 3 column layout and gutters from Page 3 but changed to left justified. The geoWrite article has 7 blank lines in the middle column on Page 4. We are going to use this space for a 'callout' later on when we add finishing touches in Page Graphics Mode. A callout helps to make the page 'less busy' and easier to read in the same way that the sub-headings on Page 3 do.

Pages 5 & 6 use 2 columns of fully justified text separated by 10 point gutters. A good example of why I don't like full justification is in the first full paragraph in Column 2 on Page 6. The line "Start geoWrite and create a new file called" has a lot of space between the words compared with the rest of that paragraph.

For Page 7 we changed to a 2 Column layout that is left justified and uses 10 Point Gutters. With the Sub-Headings and spaces between paragraphs this page has good white space and is a fairly easy read.

Now flip ahead to Page 11. Just looking at this page makes my eyes feel tired. There is a lot of text here and not much white space. I did not use any sub-headings or spacing between paragraph on purpose to show the effects of trying to cram a lot onto the page. When you combine this with the smaller font size (11 point instead of the usual 12 point) you get a very busy page. The only thing I could have done to make this worse would be to have used Full Justification which would have removed the little white space there is at the ends of lines. Page 12 uses the same small font but has spaces between the paragraphs which open the page up for a more 'eye friendly' look.

There you have it. So far we have used 7 different layouts. There are many more layouts that could be used depending on the material you are using. For the remainder of this Tutorial I will use a 2 Column, Left Justified layout with 10 Point Gutters. When necessary I'll cut down the paragraph spacing but I'll keep the pages with as much white space as I can. Continue on page 12

Continued from page 11

I will introduce one more page layout when we get to work on the cover and inside front page. The Introduction article will be laid out on 'Page i' in one full page wide column. This will give you a good look at the reason for using columns. With lines of text going all the way across the page it is hard to keep track of which line you are reading when your eyes move back across the page.

### Take Note

If you are going to use full justification make sure you don't use spaces to line up your text. Since Full Justification adjusts spacing you can't be sure your text will remain where you want it. Take advantage of the margin settings and tab stops that you can set in geoWrite and Editor mode as these won't change position when justified.

Don't forget about the Decimal Tab Stops in case you are doing a table of numbers. To create a Decimal Tab set a normal Tab Stop and then click on it with your pointer. The Tab Stop will get attached to your pointer. Press the Space Bar and the Tab will fill in and become a solid shape. Click once more to set the Tab Stop back down. When you create your table of numbers they will all line up on the decimal point.

When you are entering your text in geoWrite make sure that you leave your margins wider than the width of your columns in geoPublish. The margins will be adjusted smaller when you import your text if they are wider. If your margins are narrower they will not be expanded and you will lose some of the space in your regions.

To work around this use the ToolKit program to set your margins out wider if you have set them to one screen width for quicker typing. ToolKit makes use of margin and font settings in Text Scraps. What I have done is used Text manager to create a Text Album that contains settings I regularly use. It is then easy to make changes to documents as I work.

Next time we will begin looking at some of the many ways that geoPublish lets us import graphics into our projects.

Until then, enGEOy your Commodore!  
Bruce Thomas

## THE WRITE STUFF WORD PROCESSOR COLUMN BY: Richard Savoy

### ARTICLE NO.11

#### t,Sh-T,d      Tabs                      34

Tabs can be set/cleared anywhere within the current text width as defined by the Print Menu. Tap CTRL, then t to set/clear a tab. Before setting tabs you may wish to clear all tabs by tapping CTRL, then SHIFT-T.

Use the position indicator at the top of the screen to set tabs. C:15 means the cursor is on Column 15; T:20 means that a tab is already set at column 20; D:20 means decimal tabs are on.

Tap CTRL, d to toggle decimal tabs on and off. If you move to a decimal tab and type a number, it will be moved to the left. The cursor will remain on the tab until you press a period. This feature allows you to line up numbers on the decimal point.

#### Printing                                      35

The Print Menu is easy. You can change margin settings and several other features, but you can do a better job off or matting your text by learning to use some of the many embedded RVS format commands available to you.

Format commands allow you to access any of the special features your printer may be capable of, such as italics or underlining. You can have headers or footers automatically printed on each page. Pages can be automatically numbered.

Continue on the next page.

Continue TWS COLUMN:

You can define margins and line spacing so that your document will always print out the same way.

Be sure to read the help file "Format Commands" (file 24) for information on how to enter embedded format commands.

### Alternate Pagination 36

Use this command if you are printing on both sides of your paper and want the header or footer (usually containing a page number command) to be right-justified (aligned with the right margin) on odd-numbered pages and left-justified on even-numbered pages. This is the way pages are numbered in most books.

**he4;a#**

The above line defines a header to right-justify the page number on alternate (odd-numbered) pages only.

### Blank Lines 37

If you are double-spacing your text and want to create three blank lines, how could you do it? A carriage return creates one blank line if you are single-spacing, but two when double-spacing. The blank line command will print a given number of blank lines regardless of the line spacing used.

One limitation: This command does not work when printing in columns. Columns are usually printed single spaced, so use RETURN marks to create blank lines.

### Columns 38

You can easily print a document in double columns by setting the column margin greater than zero. The entire document will be printed in columns except for the header and footer.

To print a section of text (usually a list) in columns, use the following embedded format command:

BB Talker

BB Writer

BB Customizer

BB File Reader

The first number is the column margin, the second is the number of items in the list. The number of items must be even--if not, add a RETURN mark to the end of the list.

## THE WRITE STUFF EMBEDDED (RVS) COMMANDS

### CTRL 1 formats

**vo1** Voice (0-9)  
**sp72** Speed  
**pi64** Pitch  
**th128** Throat  
**mo128** Mouth  
**lio** Lights 0=off 1=on  
**rel** Reciter 1=on 0=off

Continued from page 15  
of January CML

## Getting Started In Machine Language

**By: Jim Butterfield**

(Article from Al Jackson 5 C's Magazine)

Programs containing machine language are a mystery to the beginning computer experimenter. It seems as if you can't list such programs and the rules for writing them are not widely known, and yet these programs often run at dazzling speed.

Machine language and its cousin, assembly language, aren't hard. But you'll need to learn new skills, such as how to deal with binary and

Continued on page 14

Continued from page 13

hexadecimal numbers. You'll need new\ tools, principally a machine language monitor program, and you'll need new information, not only about machine language itself but also about how to tap into the workings of your computer.

Several benefits come from machine language skills. You can write programs that run amazingly fast. You can do things that BASIC cannot. Perhaps most importantly, you'll get an insight into the workings of your computer.

### THE TOOLS

To begin, you'll need a machine language monitor (MLM) program. This program helps you to examine and change memory, and it even has a tiny assembler and disassembler built in. (More about these later.) If you have a Commodore 128 or the rarer Plus/4, there's a machine language monitor built into it. Just type MONITOR to get there.

A machine such as the Commodore 64 needs to have the MLM program loaded. Several such programs are available on bulletin boards and user groups. You'll find a monitor that I wrote called SuperMon in your club library.

You also need reference material: information about the instructions, about the addressing modes, and about how to call the Commodore operating system. Some of this material is provided here. You may also be able to find books that will serve as reference data.

A working knowledge of number systems, especially hexadecimal and binary, is valuable. Many machine language instructions deal directly with the bits of memory: the shifts & rotates and the logical operations. You'll understand these better if you dig into number systems.

As for hexadecimal: For the moment, you can start with the idea that the dollar sign (\$) signals a hex number and that the letters A to F are valid numeric digits. So \$AC is just as legitimate a number as 45, and \$BEAD could refer to a legal address somewhere in memory.

Bring your wits, too. You're learning new skills, and you'll need patience and attention to detail.

### REGISTERS

The working part of your (6502,09,10 micro-processor chip contains storage areas called registers. Three of these are available to hold data. These three registers are called A, X, and Y; each can hold a single byte.

The A register is sometimes called the accumulator, since the processor does much of its arithmetic there. The X and Y registers are often called index registers. The contents of index registers are often used to tweak an address, allowing an instruction to reach any one of a range of memory locations.

Most of a computer's work takes place in the registers. You bring the data in from memory, work on it, and then take it back out and store it.

### FIRST DATA INSTRUCTIONS

You can load data into A, X, or Y. The instructions are abbreviated LDA (Load A), LDX, and LDY. You can load information from anywhere in memory or just supply a value (immediate addressing). You can store data in memory, copying it from A, X, or Y. The instructions to do this are STA (Store A), STX, and STY.

The A register is a handy place for addition and subtraction. ADC is Add-with-Carry; SBC is Subtract. Values in X or Y can be bumped up or down. INX (Increment X) and INY make the register's contents one higher; DEX (Decrement X) and DEY make the contents one lower. Keep in mind that a register only holds a value that ranges from 0 to 255 (hex 00 to FF). If you go past the limit, you wrap around to the other end.

### FLOW CONTROL

A machine language program executes instructions in the order that they are stored in memory. You can change this with a Branch or Jump that takes you to a new location.

Continued on page 15

There are eight Branch instructions, all conditional. Depending on the results of a test, these instructions will cause your program to hop forward or back by up to about 128 bytes. If the test fails, the Branch won't take place, and the program continues with the next instruction. The branch instructions check flags, whose conditions are set by previous instructions.

Jump (JMP) will transfer your program's execution point to anywhere in memory.

Return from Subroutine (RTS) takes a program back to wherever it was called from. In your main machine language program, it will take you back to the BASIC program that called it.

Jump Subroutine (JSR) also takes you anywhere, but it leaves a link that allows the subroutine to return and pick up where it left off. It's like anywhere, but it leaves a link that allows the subroutine to return and pick up where it left off. It's like BASIC's GOSUB and RETURN commands. JSR is an important command for calling your computer's operating system. For example, JSR \$FFD2 asks the system to output whatever character is currently stored in A

### SYSTEM CALLS

There are dozens of system calls provided by the operating system. To begin, you need learn only two: address \$FFD2 (CHROUT) and address \$FFE4 (GETIN).

A call to \$FFD2 causes whatever is in the A register to be sent to the output, normally the screen. When you make a call to this address, you get a bonus in that the contents of all three data registers are carefully preserved. A, X, and Y will not be changed.

A call to \$FFE4 causes a character to be drawn from the input stream, normally the keyboard buffer. The character will be placed in the A register. The subroutine returns immediately.

If no character is received, A gets a value of 0. This subroutine might change the contents of all three data registers. The incoming data will arrive into A, but X and Y might change from their previous values, too.

Eventually, you can expand your catalog of calls to six, which will do almost all the work you are likely to need:

| Address | Name   | What it does                    |
|---------|--------|---------------------------------|
| \$FFC6  | CHKIN  | Switch input to logical file X  |
| \$FFC9  | CHKOUT | Switch output to logical file X |
| \$FFCC  | CLRCHN | Restore input/output defaults   |
| \$FFD2  | CHROUT | Output ASCII Character in A     |
| \$FFE1  | STOP   | Check RUN/STOP key              |
| \$FFE4  | GETIN  | Get ASCII Character into A      |

### LET'S DO IT

We can walk through the creation of a simple program, step by step. As we do so, I'll try to make holistic comments on the coding, the tools, & the computer system.

Be sure your machine language monitor is installed and ready to go. Have it active by typing MONITOR, or SYS 8, or whatever the documentation calls for. If you're using SuperMon, simply load it and type RUN.

When you start the MLM, you'll see a register display on your screen. You should recognize three registers: AC, XR, and YR.

Continue on page 16

**Machine Language Continued from page 15.**

Their contents, shown below the titles, don't matter at the moment.

Right now, our objective is to write a program that lets us touch a key on the keyboard and have that key echo many times to the screen.

We'll put this program in memory starting at address 8192, or hexadecimal \$2000. There should be memory space available there on almost any model of Commodore 8-bit computer.

### **FIRST STEP**

First, our program needs to get a key from the keyboard. Recall the subroutine call to system address \$FFE4, GETIN. The coding is JSR \$FFE4, Jump Subroutine to address \$FFE4, and we want to put it at address \$2000. To do this, enter the following line.

```
A 2000 JSR $FFE4
```

The A stands for Assemble. That is, change this mnemonic instruction into real machine code. With some monitors, such as the one built into the 128, the moment you press Return, the line will automatically change to the following.

```
A 2000 20 E4 FF JSR $FFE4
```

```
A 2003
```

The A 2000 is familiar, but the next part is new. The 20 E4 FF is the instruction as it sits in memory. Three bytes, three numbers, and they mean JSR \$FFE4. If you puzzle over it part is new. The 20 E4 FF is the instruction as it sits in memory. Three bytes, three numbers, and they mean JSR \$FFE4. If you puzzle over it awhile, you might figure out that 20 (hex, of course) must mean JSR, and the address has turned itself around. Don't worry, the computer knows what it is doing.

The computer has also written part of the next line for you. To save you from arithmetic mistakes,

it has counted off those three bytes and is printing the correct address for the next instruction, complete with A for Assemble.

Your first instruction has been placed into memory. What do we need to do next?

### **ANYBODY HOME?**

When the program runs, the first instruction asks for an input character. If there's one waiting, will be placed in the A register; if no characters are waiting, A will contain a 0 byte. Let's test for this; if A contains 0, we'll go back and try again. Here are the next two lines.

```
A 2003 CMP #$00  
A 2005 BEQ $2000
```

CMP stands for Compare (the A register). The # symbol means actual value, so the first instruction is comparing the A register against the actual value of 0.

On the next line, BEQ is Branch if Equal. It's testing the result of the comparison. If the contents of A equal 0, the program will branch back to address \$2000 and try again for an input character.

If the contents of A don't equal 0, the program has a valid character. Our next task is to print that character several times. As your screen will show you, this code will start at address \$2007.

### **START THE PRINT LOOP**

I suggest that we print the character 40 times. That's a nice round number that matches the width of the screen. If you happen to be working with an 80-character screen, you can substitute 80 if you like.

We can use X or Y to do the counting, so I'll pick X. The first thing to do is to set the counter (Load X) with a value of 0.

```
A 2007 LDX #$00
```

**Continued on page 17**

Note that once again we use the # character to say that we want the actual value 0 to be loaded, not the contents of address 0. This is called immediate mode addressing.

The character is still in A, so we can proceed to print it. The call is to CHROUT at \$FFD2. No need to worry about disturbing the count value in X; this subroutine will preserve the register values.

A 2009 JSR \$FFD2

By the time our running program gets to \$200C, the character has been printed once. Count it by adding 1 to the contents of the X register.

A 200C INX

If the count in X has not yet reach 40, we can go back and print some more. We must test X for 40 (\$28 in hex). That's a Compare-X instruction, of course, and loop back with a Branch not Equal.

A 200D CPX #\$28

A 200F BNE \$2009

Can you see why we must branch back to address \$2009 and not to \$2007 or \$200C?

As I said, that \$28 is the hexadecimal equivalent of decimal 40. Most MLM assemblers will allow you to code CPX #+40, where the + symbol indicates a decimal value. Try it and see if it works on your machine.

By the time the program gets to \$2011, it has received a character and printed it 40 times. You might like to do it again, but we must have like to do it again, but we must have a way to stop.

Let's do it this way. If the character we have just printed is not an asterisk (value \$2A), we'll go back and do it all again, But if it is an asterisk, the program will terminate with an RTS and go back to BASIC.

Here's the instruction. Compare A with value \$2A, Branch-not-Equal to \$2000.

A 2011 CMP #\$2A  
A 2013 BNE \$2000  
A 2015 RTS

That's the end of our program, but the MLM doesn't know that. It will prompt you again with a line reading A 2016. Just strike Return to cancel it.

Before you leave the MLM, you might like to look over your program one last time. Use the command D for Disassemble.

D 2000 2015

You'll see the whole program listed. If anything looks wrong, you can simply move the cursor up and type over it to correct the problem.

### BACK TO BASIC

Leave the machine language monitor by typing X. You'll see the familiar READY response after you exit the monitor.

The program is in memory, ready to go. The BASIC SYS command is similar to a Subroutine call. To activate it, use the following command.

SYS 8192

Everything will seem quiet at first, but try typing a few words on the keyboard. How's that for speed typing? This example should work, with slight variations depending on the monitor you use, on most Commodore machine

### CONCLUSION

Machine language may be new territory for you, but it doesn't have to be mysterious. Each step is logical and simple. Take a little time, give it some effort, and you too can do it.

**EDITOR'S NOTE: PLEASE LET ME KNOW IF YOU LIKE LONG ARTICLES IN A SINGLE ISSUE. Richard Savoy - Editor**

## EDITOR THIS ISSUE

This is the place I get to tell you how I put the issue on these pages together. Once I know I'll be the next editor I layout a page with margins top, sides and a one foot bottom header, then I type the bottom header with the month off that issue and the page number and save to a disk or folder used just for that issue, and make sure that I will be able to make sequential files later for the disk version of the newsletter.

Articles from the internet were Saved to a 3.5" Floppy low density 720K formatted disk, and printed using a H/P Desk Jet 940C to use as photo print Masters, I still had to get the files to The Write Stuff word processor, for the Disk version! Used Big Blue Reader on a Commodore 128, 40 column to transferred the files from a FD-2000 drive 8 to 1571 drive 9. Any files that were in Commodore format I did the reverse to print out with my printer. I hope you all are happy with my new style. The three block graphics use in this newsletter can be made larger using the **FUN GRAPHICS MACHINE (FGM)**.

Richard Savoy

**PLEASE TAKE NOTE:** On page three of the January CML is a email update for Jon Searle. He is not a current member of the group. No renewal for 2005 has been received 2-21-05

## TABLE OF CONTENTS

| <u>Page</u>     | <u>Articale</u>                       |
|-----------------|---------------------------------------|
| 1               | Meeting News <b>Linda Tanner</b>      |
| 2               | Officers, Policies, Editor Guidelines |
| 3               | The Next Editor <b>Rob Snyder</b>     |
| 3               | Free <b>Bruce Thomas</b>              |
| 4               | Expo <b>Bruce Thomas</b>              |
| 5               | Basic Speaking <b>Linda Tanner</b>    |
| 6               | Basic Speaking <b>Linda Tanner</b>    |
| 7               | Cont. Expo <b>Bruce Thomas</b>        |
| 7               | Q & A <b>Rolf Miller</b>              |
| 8               | Cont. Q & A <b>Rolf Miller</b>        |
| 8               | Everybody <b>Richard Savoy</b>        |
| 9               | Buy - Sell - Trade                    |
| 10              | Responses To Note <b>Email</b>        |
| 10              | Equipment Care                        |
| 11              | GeoPub Tutorial <b>Bruce Thomas</b>   |
| 12              | Cont. GeoPub <b>Bruce Thomas</b>      |
| 12              | TWS Column <b>Richard Savoy</b>       |
| 13              | Cont. TWS Column <b>Richard Savoy</b> |
| 13              | Machine Language <b>Richard Savoy</b> |
| 14 , 15, 16, 17 | Machine Language                      |
| 18              | Editor This Issue                     |
| 18              | Table of Contents (This list!)        |
| 18              | Wanted Disk Editors                   |

I am not able to continue the article "History Commodore Mailink Disk version" Sorry Richard Savoy

## **WANTED: EDITORS FOR THE DISK VERSION!**

We are looking for volunteers to make the back side of the "Commodore Mailink on Disk", if your not familar with it, it is a 5.25" floppy disk which usually has it's own Menu program with public domain programs, it is all in "Basic Language" or you are welcome to come up with your own way to fill the disk side with program that our members, you feel would like. Our current plan is to have a visiting editor for the July, September, & November this year. All materials can be supplies. You just have to make one side of a disk and mail to Richard Savoy. Any questions contact Richard. If you would like to do contact our president Linda Tanner