COMMODORE WORLD

THE NEWS MAGAZINE FOR COMMODORE 64 & 128 USERS

Lucky Issue 13
Volume 2, Number 8
$4.95 U.S.

HTML and the World Wide Web

Part 2 of our series on the program language

Interview with Nick Rossi

Commodore's Unreleased LCD 64

Plus...

Machine Language input
A PC keyboard for GEOS
Type-in game: Intelligence Quest

SOFTWARE SUPPORT INTERNATIONAL

We Carry America's Largest Selection of C-64/C-128 Software!

ENTERTAINMENT

10 Game Grab Bag	$9.95	Pacman	$9.97
Arkanoid 2	$7.97	Plundered Hearts	$12.97
Beyond Dark Castle	$7.97	Pool of Radiance	14.97
Beyond Zork 128	$12.97	Portal	$14.97
Defender of the Crown	$9.97	Questron 2	$9.97
Double Dragon 2	$9.97	Rampage	$7.97
Heavy Metal	$12.97	Realms of Darkness	$12.97
Heros of the Lance	$14.97	Roadwars	$7.97
Keys to Maramon	$9.97	Sidearms	$7.97
Mean Streets	$9.97	Steel Thunder	$9.97
Monopoly	$12.97	Strip Poker	$14.97
Ms Pacman	$9.97	Trump Castle Casino	$12.97
Ogre	$9.97	Wheel Fortune 1-2-3	$9.97

SPORTS & FLIGHT

4th & Inches	$7.97	Tony LaRussa Baseball	$7.97
Champshp Baseball	$7.97	Tournament Tennis	$7.97
Champn Basketball	$7.97	WWF Wrestling	$7.97
Dave Winfield Batter Up	$9.97	Acrojet	$9.97
Fast Break	$7.97	Apache Strike	$7.97
Hardball	$7.97	Blue Angels	$7.97
Jack Nicklaus Golf	$9.97	F-14 Tomcat	$9.97
Jordan vs Bird	$7.97	F-19 Stealth Fighter	$12.97
Leaderboard Golf	$9.97	Flight Sim Games	$4.97
Pro Football Facts	$9.97	High Roller	$7.97
Pro Soccer	$7.97	Jet Combat Sim	$7.97
Pure Stat Baseball	$9.97	Skyfox	$7.97
Sporting News Baseball	$9.97	Super Huey I	$7.97
Star Rank Boxing	$7.97	Top Gunner	$9.97

ACCESSORIES

C-128 RGB Cable	$17.95	Floppy Disk Notcher	$6.95
Computer Hand 2	$6.95	Dust Covers - specify	$7.95
Convert A Com	$24.95	Ergostick Joystick	$16.95
Disk Bank 10/3.5"	$2.95	CBM 1200 Baud	$9.95
Disk Bank 10/5.25"	$2.95	Aprotek 2400 Baud	$49.95
Disk Bank 100/3.5"	$12.95	1351 Smart Mouse	$44.95
Disk Bank 100/5.25"	$12.95	MW 350 Interface	$44.95
Disk Bank 70/5.25"	$7.95	64 Power Supplies	$34.95
Disk Mailers	$0.39	128 Power Supplies	$49.95
Drive Cleaners	$6.95	Printer Ribbons	CALL
Serial Cable 6ft	$8.95	Userport Expander	$24.95
Serial Cable 10 ft	$9.95	Video Ram Upgrade	$49.95
Drive Power Cable	$7.95	3.5" 10 cnt.Floppy	$7.95
User Port Cable	$15.95	5.25" 20 cnt.Floppy	$4.95
Com Modem Adapter	$15.95		

PRODUCTIVITY

1750 Super Clone	$99.95	Geos 128 v2	$44.97
Basic Compiler 64	$12.97	Graphic Label Wizard	$14.97
Big Blue Reader	$29.97	Graphics Basic	$9.97
B/W Prog Tools	$14.97	Home Designer 128	$24.97
B/W Power C	$9.97	Manager, The	$12.97
B/W Turbo Cartridge	$17.97	Maverick v5	$24.95
C128 Graphics Bundle	$29.97	Model Diet	$9.97
Christmas Model Kit	$9.97	Newsroom	$14.97
CSM Protection Man 1	$14.97	On Line Help	$9.97
CSM Protection Man 2	$19.97	Outrageous Pages	$19.97
Data Manager 2	$14.97	Postcards	$14.97
Designer's Pencil	$9.97	Printmaster Plus	$19.97
Drive Align 1541/71	$12.97	Superbase 64	$19.97
Easy Working Tri Pak	$9.97	Superscript 64/128	$14.97
Geos 64 v2	$39.97	Swiftcalc w/Sideways	$14.97

EDUCATIONAL

Early Learning Friends	$9.97	Spellicopter	$9.97
European Nations & Loc	$9.97	Stickybear Math	$9.97
Keys to Typing	$9.97	Stickybear Numbers	$9.97
Memory Manor Cart	$9.97	Typing Tutor 4	$9.97
Magic Spells	$9.97	Word Attack	$9.97
Snoopy Sky Scramble	$9.97	Word Spinner	$9.97

REFURBISHED HARDWARE

C-64 Keyboard w/ PS	$79.95	1541 Clone Drive	$39.95
C-128 Keyboard w/PS	$139.95	40 Col Monitor	$99.95
C-128D Computer	$174.95	80 Col Monitor	$139.95
1541 Disk Drive	$64.95	Printers/Call for Make	$39.95
1571 Disk Drive	$129.95	Misc.	CALL

Items Listed Above Do Not Include Shipping. U.S 48 States - Add $5.50 per order. Alaska, Hawaii & Canada - add $5.50 for the first piece and $1.00 per each additional piece per shipment. Second Day Air shipping is available. Call for shipping charges. Call Or Write For Your Free c-64/128 Catalog Listing HUNDREDS Of Commodore Products And Special Offers For Your Computer. Our Order Takers Are On Duty 6:00 a.m. - 5:00 p.m. M - F and 7:00 a.m. - 3:00 p.m. Sat. - Pacific Time.

SOFTWARE SUPPORT

Software Support Int.
2700 N.E. Andresen Rd.
Suite D-4
Vancouver, Wa 98661
(360) 695-1393

CALL TOLL FREE TODAY! **1-800-356-1179**

Major Credit Cards Accepted.

CONTENTS

LUCKY ISSUE 13 • VOLUME 2 • NUMBER 8 • FEBRUARY/MARCH 1996

COMMODORE WORLD
THE NEWS MAGAZINE FOR COMMODORE 64 & 128 USERS

GENERAL MANAGER
Charles R. Christianson

EDITOR
Doug Cotton

ASSISTANT EDITOR
Jenifer Esile

ADVERTISING SALES
Charles A. Christianson
(413) 525-0023

PHOTOGRAPHY
Wayne Wrubel

GRAPHIC ARTS
Doug Cotton
Jenifer Esile

ELECTRONIC PRE-PRESS & PRINTING
Mansir/Holden, Inc.

Cover Design by Jenifer Esile

Commodore World is published 8 times annually by Creative Micro Designs, Inc., 15 Benton Drive, East Longmeadow MA 01028-0646. Second-Class Postage Paid at East Longmeadow MA. Annual subscription rate is US$29.95 for U.S. addresses, US$35.95 for Canada or Mexico, US$45.95 for all EC Countries, and US$57.95 to all other addresses worldwide. All subscription payments must be provided in U.S. Dollars. Mail subscriptions to CW Subscriptions, c/o Creative Micro Designs, Inc., P.O. Box 646, East Longmeadow MA 01028-0646.

POSTMASTER: Send address changes to: CW Address Changes, c/o Creative Micro Designs, Inc., P.O. Box 646, East Longmeadow MA 01028-0646.

FEATURES

REVIEWS

COLUMNS

DEPARTMENTS

See Our NEW CHECKSUM Utility On Page 54

From The Editor

Welcome to Lucky Issue 13!

Here we are, we've finally arrived at Lucky Issue 13. And as you can probably see from our front cover, we even made it through the much heralded 'Blizzard of 96.' Though we are still looking for the owner of that hand. Hmmm... now that I think of it, I haven't seen our Assistant Editor lately... but I digress.

While we weigh the fact that there are no thirteenth floors in buildings against the USPS's requirement that our issues remain sequentially numbered, we can't help but feel as if we are perhaps caught in the middle of it all. But instead of worrying about it too much, we've opted to add that "Lucky" bit at the beginning—and hope for the best.

Putting our superstitions aside, we've got a pretty good line-up this issue. We're continuing our HTML tutorial, of course, which is getting deeper into what this Web-slinging script language can do. It has been our intention from the conception of this series to teach the language as well as possible within the limitations of space we have in a magazine (versus a book). Commodore users can, after all, create pretty much any of the web pages that other machines can. For that reason, we haven't limited our view of the language to just those elements that are used to create pages that viewable with a text browser (which is what Commodore users on the web are limited to at this point). What we failed to do (and didn't realize until we'd had a good tongue-lashing) is point out what HTML features you might not want to use if you have a desire to create pages for text viewing. In addition, there are also some nice tips that can be used when creating text-browsable pages. Luckily, Gaelyne Moranec has gleefully agreed to bail us out, and has supplied us with some sidebars to describe these.

Moving on, Jim Brain is back in this issue with an article about one of those Commodore computers that never came to be: The never-released LCD-64. It's a shame Commodore didn't realize the potential this computer had at the point in time that it was due to be released. Well, we've never given Commodore's marketing team any gold stars, and they probably would have wiped out any such awards with their decision not to market this product.

How about a game? A type-in game? Okay, we've finally bowed to the requests from our readers for more type-in programs. I can't say how regularly we'll do so, but this issue carries a couple. Intelligence Quest is the game I mentioned, and it's a relatively simple one. BASIC programmers who want to write their own games should find this one to be a good example of what can be done, and how to incorporate sound and graphics to make game play a little more fun.

Another type-in is our 128 utility for loading C-64 programs. It's been done before? Yeah, but not like this! The version we deliver here works with most of the different Kernal ROM modifications out there, including systems running on RAMLink. It's a good take on this subject, and includes both a BASIC loader and Assembly Language source.

And while I'm talking about type-in programs, please note that we've changed from our old Check-List program, and are now using CheckSum. Those of you familiar with the old RUN magazine will recognize this utility (though we've made a couple of cosmetic changes to it). CMD obtained the rights to CheckSum back when we bought out IDG's TechMedia division, which held the rights to all the RUN programs and articles.

By now, many of you are aware that Nick Rossi is going commercial with the next release of Novaterm. Be sure to check out the interview with Nick, then check out Carrier Detect for even more details on the new version.

This issue also marks the first appearance of David Pankhurst at the healm of BASIC Instincts. David has written for CW before, of course, but now you'll get to see him regularly.

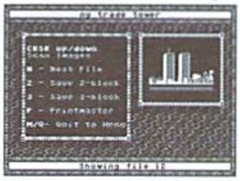
Of course, there's much, much more in this issue, but I'd hate to keep you from it any longer. Enjoy the issue, and be sure to drop us a line to let us know how we're doing!

Doug Cotton
Editor

LOADSTAR SOFTWARE VALUES!

The Compleat* Series

The Compleat PRINT SHOP I: (The Print Shop by Broderbund or Printmaster required). Over 1300 artistic and never before published PRINT SHOP images. The smart, fast software package included allows you to quickly scan through the many PRINT SHOP images sequentially, by name, or by group number. Press a key and save the graphic you want in 2-block, 3-block and even PRINTMASTER graphic files! All that plus a printed guide to your new sea of graphic files. And if you like the first volume, you'll probably want to get your hands on **The Compleat PRINT SHOP II**, which also contains over 1300 PRINT SHOP images, previously published on LOADSTAR issues over the past ten years. Included is the same smart, fast software package included in The Compleat PRINT SHOP I. Each volume is $20.00.



[illegible]

CRSR up/down
Scan Images

N - Next File
2 - Save 2-block
3 - Save 3-block
P - Printmaster
M/Q- Quit to Menu

Showing file 12

The Compleat Programmer: Become the Commodore programmer you've always wanted to be! Megabytes of knowledge crammed and stuffed onto eight 5.25-inch disks or two 1581 disks! Plus we include all the tools, extensions, languages, assemblers, tutorials and utilities you'll need to create the same type of software you see on LOADSTAR! This massive collection is way over two megabytes of instructional text and valuable tools. $20.00. For $5 more, get C= Hacking MAG, not available separately, to complete your programming set.

The Compleat Maurice: A compilation of 26 solitaire card games written by Maurice Jones, the acknowledged master of card game simulations for the C-64/128. There's even a brand new, never before published game called Boomerang. Two 5.25 inch disks or one 3.5 inch disk. $20.00 postage paid!

The Compleat Dave: Two 1581 disks or three 1541 disks crammed with SID music. Over 250 classic melodies from yesteryear, arranged and transcribed by the Master of Music, Dave Marquis, and now they're available in one gigantic 8-hour collection. If you enjoy SID music, you owe it to yourself to get THE COMPLEAT DAVE. Two 1581 disks or three 1541 disks $20.00 postage paid!

The Compleat Walt: Ten years of Walt Harned's slideshows and multimedia events gathered into one huge collection: Seven 5.25-inch disks or three 3.5 inch disks! Over 250 pictures, including some that have never been published. The greatest one-man collection of art on any computer platform! $20.00 postage paid!

The Compleat Roger: 25 educational quiz programs, each carefully crafted by Roger Norton, an educator who uses C-64s. These programs come crammed on two 5.25-inch 1541 disks or one 1581 disk. $20.00 postage paid!

*** No, we didn't misspell "complete." Compleat is the ten dollar spelling of complete.**

Other Products

Geopower Tools - 19 Geos utilities: Calendar Printer, Fast Format, Geo Fetch (grab any portion of a screen as a Photo Scrap), Phoenix (resurrect a trashcanned file), Programmer's Calculator are just a few of the handy tools. Side Two is filled with Clip Art (in Photo Album format) and fonts. $9.95 (C-64/128) Item #080525

Songsmith - LOADSTAR's own music-making program. With this deluxe music editor/player you can easily transcribe music from sheet music or make up your own tunes. Songsmith comes with a slick 30-page manual and a jukebox player with eight tunes. $9.95

Game Star #1 Eight games from LOADSTAR #70-#100). The Tenement, Stack 'Em, The Sherwood Open, Gems, Stealth Bomber, Eagle Eyes, Moonraker & Circuitry. $9.95

Just For Fun - Eight original games. There are arcade games, educational games, puzzle games and just games that are just plain fun on this disk. $9.95

Fun Four - Four original games. A huge maze game, trivia game, solitaire and a space shoot 'em up -- all runnable from a menu. $9.95

Sport - In the early days of science, "sport" meant "mutant." This is a full novel on C-64 disk by author, Jeff Jones, about a murderous neo-human. Over 500 terrifying pages. Automatic presentation software included. Printing capability. Bookmarks. Warning! This is a *real* novel with strong content, frank language, violence and adult situations. If the hard-hitting action of *real* sci-fi/horror novels turns your stomach, please don't buy this book. Must be 18. 1 CMD HD disk , 2 3.5-inch disks or 3 5.25-inch disks. $5.95 $1.00 Shipping.

Brainpower/Brainstorm - 80-column word processor and idea processor for the C-128. These two programs are together on one disk. It comes with a detailed 32-page manual. $9.95 (C-128 only) Item #069421

Best Of Loadstar Compilations: LOADSTAR is over 11 years young! We have 5 anthology disks that take you through the evolution of LOADSTAR, from humble beginnings to the well-oiled machine we've become. If you want to hop on a time machine and go back, start with **The Best Of LOADSTAR #5** and move your way back to **The Best Of LOADSTAR #1.** Each of the 136 back issues are available for purchase with discounts on large orders for you collectors out there. Best Of LOADSTAR disks are available on 5.25-inch disks only. C-64 disk. $9.95.

Master Base - Database of users' groups for the 80-column C-128. This is a fast, powerful database program for handling addresses and mailing labels (includes barcode printing). The disk also has a file of over 600 users' groups addresses. $4.95 (C-128 only)

LOADSTAR is a monthly "magazine on disk" for the Commodore 64/128. Subscribers receive two 1541 disks (or one 1581 disk) in their mailbox every month filled with news, articles and programs. These non-PD, high-quality programs are written by the best home-based programmers in the field and edited by the crack LOADSTAR team of Fender Tucker and Jeff Jones. Subscription prices are at an all-time low of $69.95 for a 12-month subscription, or $19.95 for a three-month subscription. You may also elect to subscribe "by the month," where we charge your credit card $7.95 for each issue after it's shipped.

Loadstar Products Order form 1-800-594-3370

Quantity	Description	Price	Item#	Total
	3-month LOADSTAR subscription	$19.95	na	
	1-year subscription to LOADSTAR	$69.95	na	
	Month to month @$7.95	$7.95/mo	na	
	Compleat Print Shop I on 5.25-inch	$20.00	0009D5	
	Compleat Print Shop I on 3.5-inch	$20.00	0001D3	
	Compleat Print Shop II on 5.25-inch	$20.00	0010D5	
	Compleat Print Shop II on 3.5-inch	$20.00	0002D3	
	Compleat Programmer on 5.25-inch	$20.00	0005D5	
	Compleat Programmer on 3.5-inch	$20.00	0005D3	
	Hacking mag 3.5-inch only	$5.00	0006D3	
	Compleat Maurice 5.25-inch	$20.00	0007D5	
	Compleat Maurice 3.5-inch	$20.00	0007D3	
	Compleat Dave 5.25-inch	$20.00	070525	
	Compleat Dave 3.5-inch	$20.00	070523	
	Compleat Walt 5.25-inch	$20.00	070425	
	Compleat Walt 3.5-inch	$20.00	070423	
	Compleat Roger 5.25-inch	$20.00	0004D5	
	Compleat Roger 3.5-inch	$20.00	0004D3	
	Geopower Tools	$9.95	080525	
	Game Star #1	$9.95	080825	
	Just For Fun	$9.95	073525	
	Fun Four	$9.95	073525	
	Sport 5.25-inch	$5.95	070325	
	Sport 3.5-inch	$5.95	070323	
	Sport CMD High Density	$5.95	070327	
TOTAL FROM THIS COLUMN				

Qty	Description	Price	Item#	Total
TOTAL FROM LEFT COLUMN →				
	Songsmith	$9.95	069525	
	Best Of LOADSTAR #5	$9.95	049525	
	Best Of LOADSTAR #4	$9.95	049425	
	Best Of LOADSTAR #3	$9.95	049325	
	Best Of LOADSTAR #2	$9.95	049225	
	Best Of LOADSTAR #1	$9.95	049125	
	Masterbase	$4.95	081025	
SUBTOTAL				
Louisiana Residents ad 4% sales tax				
Shipping $5.50 UPS Second day air. Pay no shipping if LOADSTAR subscription or any product from the Compleat series is ordered.				
Total Enclosed:				

Payment method:

❑ Check/money order made payable to "Softdisk Publishing" in US Funds

❑ MasterCard ❑ Visa ❑ American Express ❑ Discover

Card #____________________________

Expiration Date _____/_____/_____

Authorized signature: ________________________

Subscribers: I prefer: ❑ 5.25-inch diskettes ❑ 3.5-inch diskettes

Mail or Fax to:
Softdisk Publishing
P.O. Box 30008, Shreveport, LA 71130-0008
Questions: ○ 1-318-221-8718 ○ Orders 1-800-594-3370

Commodore Trivia

by Jim Brain

Welcome to another edition of Commodore Trivia. As many of you may know, these trivia questions and answers have been donated by me to the Commodore community at large. Unlike other articles in *Commodore World,* these trivia questions have been placed in the public domain. I ask only that the trivia questions remain intact and unchanged, and that my name and address appear somewhere so users can contact me. The trivia is also used for a contest I run on the Internet; contact me at the included address for more information. Because curiosity has the best of me, I always welcome a note or postcard detailing where the trivia goes. I also welcome new questions—provided they come with the answers. Enjoy!

Jim Brain
Brain Innovations, Inc.
602 North Lemen
Fenton, MI 48430
brain@mail.msen.com

COMMODORE TRIVIA #12 QUESTIONS

$0B0 What will happen if you type ?""+-0 into the CBM BASIC interpreter on the PET series, the 64 series, or the 128 series?

$0B1 In the first CBM 64 units, what color was the screen color RAM changed to when you cleared the screen?

$0B2 Why was it changed in later versions of the 64?

$0B3 What is "special" about the text that displays the "illegal quantity error" in CBM BASIC?

$0B4 On what Commodore machine was the operating system OS/9 available?

$0B5 Which Commodore machine(s) does not have a user port?

$0B6 How many pins are there in a Commodore Serial Connector?

$0B7 There are 13 addressing modes available on the 6502. Name them.

$0B8 If you were to put one large sequential file onto an 8050 disk drive, how big could that file be?

$0B9 How many characters can be present in a standard Commodore DOS filename?

$0BA How many pins does a 6502 IC have on it?

$0BB How many pins does the standard IEEE-488 connector have on it?

$0BC On the IEEE-488 bus, what does the following acronym for pin 7: NRFD, stand for?

$0BD On the NMOS 6502, what is the ML opcode for SED, and what does this opcode do?

$0BE Assuming that a PET computer and a non-PET computer have access to a common disk drive or tape drive, there are two ways to load a PET BASIC program on the non PET CBM computer. Name them.

$0BF Only one of the ways detailed in $0BE works the other way around. Which one?

COMMODORE TRIVIA #11 ANSWERS

$0A0 The 1700 (128kB), the 1764 (256kB), and the 1750 (512kB). The 1700 and the 1750 were marketed for the 128, while the 1764 was marketed for the 64 line.

$0A1 The only one I have documented in front of me is the timer B interrupt bug, which is explained in the "Toward 2400" article by George Hug in Transactor 9.3. (1) However, I had many people relate other bugs (2 and 3), which I haven't been able to test, so I add them as possibilities. (I encourage readers to confirm or deny the latter two.)

1) If timer B of the 6526 CIA times out at about the same time as a read of the interrupt register, the timer B flag

may not be set at all, and no interrupt will occur if timer B interrupts were turned on.

2) When the hour on the TOD clock is 12, the AM/PM must be reversed from its normal setting to set/reset the AM/PM flag.

3) The TOD clock sometimes generates double interrupts for alarm trigger.

$0A2 I was looking for the B series computers, which contains the B computers (B128, B256), as well as the 600 series and the 700 series. These computers could be fitted with an optional 8088 processor on a separate card. However, another correct answer is the Amiga, which can have a 8088 attached via an expansion card or a SideCar(tm) unit.

$0A3 1953.125Hz, 2840.909Hz, and 1488.095Hz. These correspond to waveforms with periods: 512us, 352us, and 672us, respectively.

$0A4 The tones recorded on the Plus/4-C16 are exactly one-half the frequencies shown above. This suggests to many that the Plus/4 and C16 were supposed to run at twice its present frequency, but were downgraded at the last-minute, and the code to generate the tones was not updated to reflect the change. This is just hearsay, so you decide for yourself.

$0A5 It sets the raster compare interrupt to go off at scan line 311. If the interrupt occurs, we are on a PAL system, since NTSC will never get to line 311 (NTSC only has 262.5 lines per frame, every other frame shifted down a bit to create 525 lines).

$0A6 $00, or 00

$0A7 The program counter gets saved high byte first, then the processor status flags get saved.

$0A8 $0100 to $01FF

$0A9 The overflow flag: V. However, the V flag can be set via an external pin on some members of the 65XX line. The 1541 uses this as an ingenious synchronization tool.

$0AA Some terminology: "inter" means "between". Most everyone knows that a tape block is recorded twice on the tape, but Commodore considers the two copies and the gap between them a single "record". Thus, this question is referring to the gap in between two dissimilar records. With that in mind, the interrecord gap is nominally 2 seconds long, (or 223.2 byte lengths, although the gap contains no data). It is there to allow the tape motors to get up to speed before the next data comes under the read/write head. The tape motors may need to stop between records if the program is not requesting any more data from the tape data file at this time. If the program subsequently asks for data from the tape, the drive must get up to speed before the read can occur. Note: on the first version of PET BASIC, the gap was too small, so programmers had problems retrieving data files.

For completeness, the "intra-record" gap (the one between the two copies of the data) consists of 50+ short pulses, each of which is 352us in length, giving a timing of .0176s+. This time was used to copy important data to safe locations, reset pointers, and do error logging. The entire "record" is recorded in 5.7 seconds.

$0AB $1e00, or 7680

$0AC lO (L SHIFT-O)

$0AD lI (L SHIFT-I)

$0AE $c000, or 49152

$0AF The text "CBMBASIC" is stored there. It is strange because this text is not referenced by any routine. It can also be called strange because the code is Microsoft's. Doesn't it make you wonder?

ON THE HORIZON

COMMODORE AND COMPUTER INDUSTRY NEWS

CMD SuperCPU Update

Word is that things are moving smoothly, and CMD appears to be on schedule for release of the SuperCPU 64 20 MHz accelerator, slated to release sometime in April of this year. The initial hardware prototype has now undergone preliminary testing stages, and CMD is proceeding with circuit board and case design phases of the project.

CMD has also announced a preliminary price of $299 for the SuperCPU 128, slated for delivery in the third quarter of this year. Deposits of $50 for this version are currently being accepted which guarantees the depositor a first-run unit at a price no higher than the preliminary announced price. Also note that CMD will transfer deposits for the 64 version to the 128 version, but urges anyone wishing to do this to inform them quickly to avoid being shipped a 64 version.

Some additional information concerning the SuperCPU 64 has been provided by CMD, including some results of initial tests.

First, many users have asked about using a SuperCPU 64 on a C-128 computer in 64 mode. CMD has stated from the beginning that this configuration would work, and testing of the prototype has confirmed that it does. Next, CMD had initially stated that extra RAM for the SuperCPU might be usable as a fast disk. Some users have speculated that this would then supersede RAMLink. Such an option would not fully supersede RAMLink, however, since the SuperCPU RAM would lack the power backup to retain such memory when the computer is off. Furthermore, CMD has decided not to pursue the fast disk option, citing that the required development time would not be available in the near future.

As for some of the test results, CMD has run some early experiments to test the compatibility of the unit, and to get some rough estimates of the amount of speed enhancement that could be achieved. One simple BASIC FOR/NEXT loop executed over 21 times faster than on a stock system. CMD explains that while the CPU runs only 20 times faster, it can achieve higher rates of efficiency by using reduced memory mirroring and due to not having to be interrupted by the VIC (the VIC gets its data from RAM inside the computer).

CMD tested a number of commercial games as well, and found that many of them ran at normal speed with the accelerator set for 20 MHz. This is obviously attributable to interrupt-based event-timing used by game programmers.

Operation of Commodore REUs with the SuperCPU at high speed has also been confirmed. Commodore's REU demo programs (the globe and pound sign demos) ran blazingly fast. CMD also ran timing tests on a new demo that is currently making its way around Commodore circles, called Polygonamy. The demo rotates 3-dimensional objects in real-time using some very fast routines, and achieved a frame rate of between 12 and 13 frames per second on a stock machine. Without any optimization, the demo turned in a rate of approximately 128 frames per second using the SuperCPU. CMD says that an optimized version could probably achieve 200 frames per second or more.

GEnie Sold, Price Restructuring Announced

With a flurry of press releases, new management at Genie (formerly GEnie) recently shocked subscribers with an immediate price restructuring plan that favors those who make heavier use of the system—and punishes those who do not.

Genie has housed the largest Commodore 64/128 support library ever since the demise of Q-Link. Because of this, and because CMD and CW have maintained a regular support area on Genie for several years, we feel obligated to publish the full range of releases—even though this will take the majority of our news column for this issue (it was a slow news month, anyway). We do this so that users can judge for themselves the merit of the changes taking place.

I do wish to point out that CMD and CW will continue to provide and maintain support areas on Genie until such time as it is no longer reasonable to do so. Only time will tell if the new price structure will negatively impact the Commodore community there to the point that we no longer find it a viable avenue for providing support.

Without further ado, here are the press releases:

Genie Price Restructuring Announced

The new Genie is here ... and you can throw any concept you've had about "value" out the window because we're about to redefine it for you!

Define value. OK. How about unlimited 24 hour-a-day access to the Internet for one flat price??? How about surfing the Worldwide Web or posting to Usenet newsgroups anytime night or day without having to watch that billing clock?

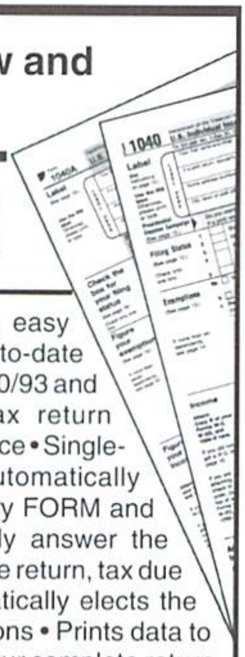
The new Genie is bringing you that and more! Here's the scoop...

As of February 1, 1996 for just $29 a month you can get unlimited SLIP/PPP connections in 47 states! Plus, when you subscribe to Genie Internet Service you'll pay no monthly subscription fee for regular Genie Services! What's more, when you access your regular Genie Online Services account, the hourly charges are only $2.75 per hour!! Finally (no, we're not done yet), the surcharge for weekday access during Business Hours will be reduced from $2 per hour to only $1!!!!!

...continued on page 8

PERFECT TAX™

Get PERFECT TAX™ now and relax on April 15th...



- **THIRTEENTH SUCCESSFUL YEAR! • MANY THOUSANDS ALREADY IN USE!**
- **Your Simple Answer to the TAX Reform Acts of 1986 through 1994**

- Stores data files on disk for quick recall
- Yearly upgrades available to registered users
- Fully screen-prompted, menu-driven and easy-to-use
- Includes helpful tutorial example and forms library
- Built-in calculator accumulates input and enters totals
- Instant recalculation of your entire return when you change any item with just a single keystroke
- Prints directly onto IRS forms and Schedules or to blank computer paper for use with transparent overlays
- PERFECT TAX is Tax Deductible

Supports all of these FORMS and Schedules:
FORMS: 1040, 1040A, 2441, 3903, 6251, 8615 and 8815
Schedules: A, B, C, D, E, SE, 1, 2, 3 and EIC-A

TAXPERFECT 128 additionally supports:
FORMS 2106, 2119, 2210, 3800, 4136, 4256, 4562, 4684, 4952 and 8396, plus Schedules EIC, F and R

• PERFECT TAX is the quick, correct, easy way to do your taxes • Absolutely up-to-date with the Tax Reform Act of 1986/87/88/90/93 and *all new* tax changes • Simplest tax return preparation program available - at any price • Single-keystroke form-to-form change • Automatically calculates and transfers data from every FORM and Schedule to the FORM 1040 • Simply answer the questions - PERFECT TAX calculates the return, tax due or amount of refund due you • Automatically elects the greater of Standard or Itemized deductions • Prints data to all FORMS or Schedules you need for your complete return - ready to sign and file • Highly acclaimed by the tax pros, CPA's and tax preparers, PERFECT TAX is easy to understand and a pleasure to work with • Separate versions available for Commodore 64 and Commodore 128 computers

PERFECT TAX 64	**$69.00**
PERFECT TAX 128	**$79.00**

(plus shipping—see our main ad)

Available from
Creative Micro Designs, Inc.
1-800-638-3263

FOR THE BEST SELECTION OF
REFURBISHED COMMODORE EQUIPMENT
CALL 1-800-638-3263

JiffyDOS pre-installed on every computer & disk drive!

30 DAY WARRANTY on all refurbished equipment

COMPUTERS	
C64 or C64c (refurbished)	$89.00
C128 (refurbished)	$149.00
128-D (refurbished)	$239.00
MONITORS	
1702 (refurbished)	$99.00
1802 (refurbished)	$129.00
80-column monitors	CALL
DRIVES	
1541 (refurbished)	$75.00
1541-II (refurbished) ***SPECIAL!***	**$89.00**
1571 (refurbished) ***SPECIAL!***	**$99.00**

Prices do not include shipping charges, and are subject to change without notice. All items subject to availability, call before ordering.

SPECIAL! 1541-II Drive **$89.00**

SPECIAL! 1571 Drive **$99.00**

CMD **Creative Micro Designs, Inc.**

ON THE HORIZON

Commodore and Computer Industry News

...continued from page 6

Subscribing to Genie Internet Service has other benefits yet to come. Genie will be moving to the Internet over the coming months. Your Genie Internet Services subscription will get you reduced rates and, in some cases, FREE access to our new Internet-based services. We'll tell you more about these services and benefits as soon as we can.

Of course, if you don't want unlimited access to the Information Superhighway, if you want to be left in everyone's technological dust, we've still got ways for you to save!

Starting February 1, 1996 we're giving you MORE FREE HOURS ... FIVE more hours for a total of NINE every month! And we've lowered the charge for additional hours to only $2.75!!! Plus we've reduced the weekday Business Hours surcharge to only $1 per hour. All this for the low monthly charge of $23.95!

Oops ... did we say $23.95? That's what new Genie subscribers will pay. But for all of our existing members, we're even reducing that to only $18.95 a month!!! And the reduced rate of $18.95 will be good for as long as you keep your Genie Online Services account!

How's THAT for defining value?!?! More FREE hours! Unlimited Internet Access! Reduced weekday Business Hours rates! We're excited about the new Genie, and we're sure you will be, too!!!

PS - To subscribe to Genie Internet Services and get unlimited SLIP/PPP access for only $29 a month, either call Genie Client Services at 1-800-638-9636 or send electronic mail to FEEDBACK.

New Canadian Pricing Finalized

New pricing has been finalized for GEnie's Canadian subscribers. Effective February 1, 1996 all charges from Genie will be made in U.S. Dollars at the same rates charged to U.S. subscribers. Those rates are:

New Subscriber:	$23.95	Monthly subscription + 9 hours connect time
	2.75	Charge per hour after 9 basic hours
	1.00	Surcharge per daytime "prime time" hour
	3.75	Surcharge per hour for Datapac access
Current Subscriber:	$18.95	Monthly subscription + 9 hours connect time
	2.75	Charge per hour after 9 basic hours
	1.00	Surcharge per daytime "prime time" hour
	3.75	Surcharge per hour for Datapac access

Plus ...

Coming soon to a Province near you ... direct, unrestricted access to the Internet for one low monthly fee. Our Internet provider is now, as this is being written, working to finalize their access capabilities in Canada. More details to come as service becomes available.

A Letter From The President

Dear Subscriber:

I regret if our unexpected price restructuring caused anyone confusion. By the time you're done reading this message I sincerely hope that you'll have a clearer understanding of our vision for and commitment to the future of Genie.

We have no intention of hanging Genie out to dry or allowing it to become any more obsolete than it had before we arrived on the scene. We agree with the comments many of you have made. Genie is a great service. In fact, some aspects of our service are unequaled anywhere else in the on-line services industry. We recognize, though, (and we hope you do, too) that the only way to survive is to dramatically improve our infrastructure and plunge head-first into the 21st century.

For our members who use Genie frequently, this price structure will result in savings. Just do the math. Genie members who access the service for as little as 5 hours a month are saving money. Here's an example:

Old Plan: 6 hours of usage		New Plan: 6 hours of usage	
4 hours	= 8.95	9 hours	= 18.95
2 hours x $3	= 6.00	6 hours prime time x(1.00)	= 6.00
6 hours prime time x $2.00	= 12.00		
Total	= 26.95	Total	= 24.95

This first example assumes that your 6 hours will be used during prime time. I realize that isn't the case for many of you, so now let's eliminate the prime time usage from the equation. Your savings will be slightly delayed, but you would still start saving at just over 7 hours of use a month. Here's another example:

Old Plan: 8 hours of usage		New Plan: 8 hours of usage	
4 hours	= 8.95	8 hours (1 hour left over)	= 18.95
4 hours x $3	= 12.00		
Total	= 20.95	Total	= 18.95

Any individual who uses Genie more than 5 hours a month during prime time, or just over 7 hours in non-prime time, will save under the new rate structure. This is the most important part of my message. We expect new capabilities. We expect faster access. We expect better games and entertainment services. We expect more reliable service and connections. With these expectations comes the realization that our members will want to spend more time on-line, too. We are firmly committed to providing these capabilities to our loyal customers at a price which makes them more affordable than our old price.

There is much more in store for Genie users. It is extremely difficult for an on-line service to grow and provide the best for their customers without offering full, graphical access to the hottest topic of the decade. The INTERNET. But not just access, complete and total access. Access to every aspect and every area of information the Internet has to offer. We are offering our customers, both old and new, the ability to access the full Internet without any restrictions or limitations whatsoever.

And our GenieNet service gives you true SLIP/PPP connections. Unlike other internet services, we fully support applications such as RealAudio, WebPhone, and full mpeg video. You'll also get the best, fastest, and most user-friendly software available to navigate the Worldwide Web ... Netscape of course. You'll be accessing the Internet through a different network than your regular Genie access, and nearly every line of the Internet access network offers connection speeds up to 28.8 kbps.

You'll have a worldwide E-mail address to send unlimited e-mail and unrestricted access to over 14,000 newsgroups and BBS's through Usenet. You get FTP, GOPHER, ARCHIE, WAIS, VERONICA, FINGER, PING, IRC chat lines, as well as full access to the WORLDWIDE WEB. Plus you'll get storage space. Lots of storage space. Up to 8 megabytes of storage space on our servers. Absolutely FREE. Plus you can put up your own complete WEB SITE or just a simple home page.

All this, a solid network, the very best WWW browser for free, loads of capabilities on the Internet ... plus the added benefit of having access to the finest gaming and the best RoundTables any on-line service could offer. All for a flat rate of $29.00. The GenieNet account gives you the best of both worlds. Unlimited, unrestricted access to the Internet at $29.00, and your monthly fee to access Genie is eliminated.

You pay only for the time you actually use ... at the newly reduced rate of $2.75 per hour (and reduced prime time surcharge if you access during weekday business hours). This power package will blow away any Internet access you may have on any other service.

I trust I've done a better job of explaining the situation, and that you now understand the direction we're headed. We'd like you to stay a part of us, and I hope you'll give our Internet access a try, too. There is no set-up fee, no connection charge, and no contractual obligation. The Netscape software is yours to keep. We believe you should have had these capabilities a long time ago, and we're going to do our best to keep bringing you opportunities from the cutting edge.

Thank you again for your loyal support of Genie. Be sure to enjoy the new and always improving GenieNet!

Sincerely,
Geoffrey Rochwarger, President
Yovelle Renaissance Corporation

Important Notice About Changes in Billing Procedures!

Genie is changing the way it bills you for monthly fees and usage charges. In the past, your monthly subscription fee and any extra time you used has been charged to your designated method of payment in the following month.

In February your designated method of payment will be charged for your January subscription fee and any usage or other fees you incurred during January. These charges will be posted to your designated method of payment around February 17th. Your February subscription fee of $23.95* will be charged to your designated method of payment sometime after the January fees and usage have been charged. This means you will see two charges from Genie to your designated method of payment in February.

In the future your subscription fee will be posted to your Genie account by the first of the month. It will be billed, combined with any usage or other fees from the previous month, to your designated method of payment in that same month.

Specifically, your March fee will be posted to your Genie account by March 1. Within 1-3 weeks it will be posted to your designated method of payment and combined with any usage charges or other fees which may have been posted to your Genie account during February. After March we will continue the policy of charging your designated method of payment for the current month's subscription fee combined with any additional usage or fees from the previous month.

* Subscription fee is only $18.95 for members who initiated their Genie subscription prior to February 1, 1996

Additional Changes effective March 1, 1996:

Effective March 1, 1996, Genie will implement a $10 processing fee for Insufficient Funds and Over the Limit returns from your method of payment.

If you have any questions about these changes, please call Genie customer service at 1-800-638-9636.

Just For Starters

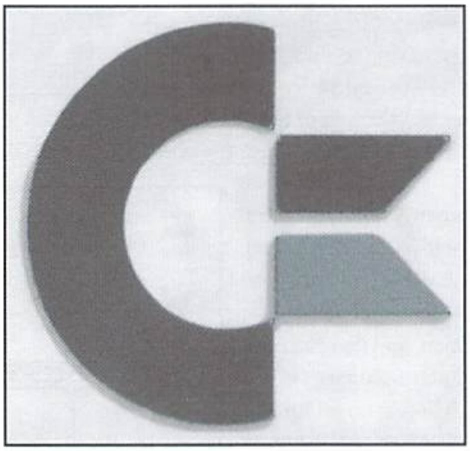
by Steve Vander Ark

THE WINDOWS 95 CONNECTION

I spent two hours today messing around with Windows 95. I don't know if you've had a chance to try this latest behemoth of a program from the World's Mightiest Software Company. If you haven't, you really should try it sometime. Basically, it's GEOS "all grown up", by which I mean that it acts an awful lot like GEOS for the IBM/PC, which in turn acts an awful lot like GEOS for the Commodore. True, each incarnation of the program adds a heck of a lot more power and features, but deep down, they all have the same heart. When you start using them, you can feel it and understand it. Simply put, Windows 95 is cool. And it's neat to see its heritage traced right back the Commodore.

But in my opinion, what is even more cool is that when I finished messing around with Windows 95, the company whose computer it was wrote me a check for my time and expertise. Yes, I got paid for messing around with a computer. That's pretty cool! I also get paid for writing this column, and that's cool too. It's also a little bit mind-boggling. I can hardly believe that I am considered to be an expert, or at least knowledgeable enough for someone to give me money for what I know. But that's what happens; and quite frequently, I might add.

Now, I'm not bringing this up to proclaim to the world that I'm an expert. As a matter of fact, I don't really even consider myself to be an expert, since I know quite a few people who really are, and they know a lot more than I do. I consider myself to be just a glorified beginner, someone who spends a lot more time learning than anything else. That puts me in exactly the same boat as you folks who read this column. We're all learning together.



So, you ask, how come this company paid me to poke around in their computer, setting up Windows 95 the way they wanted it? Were they trying to tap into some vast storehouse of computer knowledge in my brain? I hope not, otherwise they were ripped off. Did they think I was a Windows 95 guru who had perhaps been one of the beta testers and who knew the program inside and out? Once again, I hope not. I have read a book or two on Windows 95 and I do try to keep up with the Windows 95 RoundTable on Genie, but I am no wizard with that operating system. So, what was it, then?

This was what I was mulling over as I started planning this column. Why me? What did I have to offer people? Why do I write columns about computer subjects? Why do I get called upon to fix a program that won't run? What is it that I know that serves me in those situations? The answer is not immense knowledge or a genius for computer code. What it is, really, is a specialized kind of common sense (that I have no name for) but which makes all the difference in the world when you plop yourself down in front of a keyboard.

This kind of common sense is what makes a computer user into an expert; a student into a teacher. It means that you know what to expect, how things work, and, most importantly, where to find out what you don't know. So, why talk about that in this beginner's column? Because that kind of common sense is exactly what each of you is trying to develop as you spend time working on your Commodore computer. (And don't you think for a second that you need a Pentium to gain this kind of experience... I learned most of what I know on the Commodore and you'll find that a lot of veteran computer users did too.) It's the reason you know enough to hit the return key when a program loads a fancy introduction screen and you want to scoot past it into the game itself. It's the reason you know what to do when a program doesn't load properly when you type LOAD"*",8,1 and press RETURN.

Take these first few columns, for example. I wrote about the way that your computer talks to you and to the various gadgets you have hooked up to it. I mentioned error messages. I also described something about how your computer routes stuff to the correct place; whether that was to your monitor screen, a disk drive, or whatever. All of that information helps you to begin to understand what's going on and therefore be able to figure out what you're supposed to do next. As you use your computer, those underlying ideas become instinctive. You don't think twice about error messages anymore. You just use what they tell you to make it possible to keep working. In other words, you gain a little of that common sense about computers that could eventually make people think you're the expert.

Companies have capitalized on the whole concept of building common sense as a way to make computers easier to use. The Macintosh operating system, Windows, and GEOS are all examples of software which takes advantage of this idea. They do it by setting up a whole series of rules for how their program works and demand that anyone writing programs to go with it follow those same rules. It works extremely well. Every Windows application displays a series of menu choices across the top, for example, and the first one is always "file." In any of these programs, no matter which company created them, you will be able to open, close, print, or save your document from this menu. And the way you do those tasks; the windows that open and the buttons you have to push, is always identical as well. That's how I can sit down in front of a computer and install, set up, and run a program I've never seen before without looking at the manual for instructions.

There isn't any such set of rules for Commodore programs outside of GEOS. But there are still plenty of assumptions that go into every Commodore program, things like the error messages and disk commands. As you read this column, I hope you will learn some of those assumptions.

They taught me in college when I was getting my teaching certificate that you can really help people out by giving them something called an "advance organizer." What that means is that you give them some idea of what they're expected to learn. You can hand out outlines or diagrams; get their curiosity piqued with some example, or just lay it on the line ("See this word? It's pronounced 'parallel,' and today you're going to find out what that means.") What I've been talking about in this issue so far had really been just that: an advance organizer for you, the reader. Every issue I will spend my allotted word count discussing disk drives, user ports, error messages, and all those other little details that make up Commodore computing. As you read, keep in mind that I am hoping you can build your computer common sense. I'd like you to start to feel more and more like an expert as you internalize some of what your computer is doing and understand a little better what it expects you to do in return.

And who knows? Maybe in a few years you'll be the one writing this column! Wait a minute, on second thought...

Back to BASIC

As you get more and more into programming, you will find that there are a few utility commands and procedures you'll need to know. Before I get back into loops and input routines, let's talk about some of these.

First of all, you need to get comfortable with the LIST command. You need it because it's how you get your program code to appear on the screen. If you write some lines to create a graphics screen like we talked about last issue, then press RETURN, the BASIC lines you wrote will very likely have vanished from the screen. In order to see them again, you need to use the LIST command. You type this on a blank line, and if you just type LIST, you'll see your whole program listed for you. From there you can cursor up into the program and make changes to your heart's content. (The changes won't actually be saved, however, until you press RETURN on the line you changed.)

LIST has a few options to it which come in very handy. If you follow the LIST command with a line number, for example, only that line will be listed. You can also specify a range of line numbers to be listed. Let's say you wanted to see all the lines from number 10 through number 50. You'd type LIST 10-50 and hit RETURN. Those line numbers would appear.

Once you start writing more and more complicated BASIC code, you will find yourself needing to add lines. Since BASIC uses numbered lines, you will have to have a line number available where you need a new line to go. You will remember that we have been numbering our program lines by tens. This was intended to leave that extra space just in case. If you have a quite a few lines to add, though, even ten line numbers can be too few.

One way around this is to place more than one BASIC statement in a numbered line. It's actually very simple to do this. You simply have to separate the commands with a colon. Here's an example:

These five lines of BASIC...

```
10 PRINT "HI!"
20 POKE 646,5
30 POKE 53280,7
40 PRINT
50 PRINT "HOW ARE YOU?"
```

...can be converted to just one line like this:

```
10 PRINT"HI":POKE646,5:POKE
53280,7:PRINT:PRINT"HOW ARE
YOU?"
```

That can free up quite a lot of space, as you might imagine. The only limitations on this kind of thing is that no program line can fill more than two lines of text on the screen, which on the Commodore 64 means a limit of 80 characters.

Another way to gain space is to use a GOTO command to jump the program to another line where there is enough space to create the lines you need. Then you just GOTO back again to where you left off.

As you might imagine, programs that use these tricks can get pretty complicated. Trying to follow the logic of such a program can be a daunting task. And when errors appear, as they almost certainly will, a program which has commands all crammed together and odd jumps off into some other corner of the program is much harder to debug. Programmers refer to these kinds programs as "spaghetti code." It's a habit worth avoiding at all costs.

Instead, plan ahead. Use more than one command on a line only when it makes sense to do so; when you want to PRINT several times in a row, for example, or as part of an IF-THEN loop. Write your program in sections and assign each section a range of line numbers. You might save lines 10 through 200 for your opening screen and starting input routines, then start at line 300 for the next section. All those unused line numbers can be used to expand a section without bumping into the next one. You'll find that a well-planned program is not only easier to fix or modify but also more gratifying to write in the first place.

Here's a neat trick that can come to the rescue if you need to move whole lines to new line numbers. LIST the lines you want to move on the screen. Now cursor up to the line number of one of the lines, DEL the number, and type in the new number you want it to have, and press RETURN. You have now copied that line to a new line number. You do need to go in and get rid of the old one, however, but that's easy to do too. Just type the old line number on a blank line and press RETURN. That line is now empty.

Foreign Exchange

By Joseph Gaudl

THE GERMAN GEOS USER INTERFACE

Just about the same time Maurice Randall's geoShell hit the market in the States, a very similar program was causing quite a stir among GEOS users in Europe. Jurgen Eckel was offering with his new program "CLI", a text-based MS-DOS type interface for GEOS. Jurgen hit pay dirt with CLI, as it has outsold and continues to outsell geoShell by a long shot. Most Europeans are used to using programs that are written in English. That's the case regardless of the computer platform. If the program is good and does what they want, they use it. They might pronounce the English words all wrong, but hey, they're getting their work done. CLI is written in German and English but the real reason for its success is that it has more to offer than geoShell. The program is exceptionally well written and the latest version (2.5) and it just doesn't get any better than this.

CLI is the abbreviation for "Command Line Interpreter", and that basically explains what the program does. Instead of using the graphic deskTop with icons, the user simply types in commands with the keyboard, presses the RETURN key and the interface escorts you to your application or drive or carries out different commands. Bypassing the graphic environment makes working with GEOS extremely comfortable and speeds up everything considerably.

All types of drives and RAM expansions can be accessed, including CMDs mass storage devices. Emulation modes, native modes and subdirectories are no problem for CLI. The program is only 20 kbytes and resides in the computers memory at all times. No part of the program needs to be reloaded—ever.

The commands for CLI were "adopted" from the world of MS-DOS. Those of you who also use a PC (shame on you!) will recognize practically all of the commands in CLI. The rest of you will have to do a bit of learning the ropes, but will catch on to the process pretty quickly. A batch-file editor makes the program even more like a IBM compatible machine and means faster access and greater comfort. The F-keys are defaulted to issue commands when pressed (16 commands possible!) and can be redefined by the user.

The installation is quick and easy, but there is one tricky part. After the installation has started, a dialog box appears asking for one of fifty codes that can be found in the handbook. Use the proper code! Each installation can only be made once. You blow it and you've just wasted 20 German Marks! If the installation is successful, and it should be, the program allows you to protect your entire system with something I just hate: a password. (I use the same password for everything that requires one so I don't forget it. If I should forget it remind me that its "Yodeler".) The default password is—you guessed it—GEOS.

When you boot your system the next time, a dialog box appears asking you if you want to enter the deskTop or CLI. Choosing CLI will cause the drive prompt "A:" to appear. The rest is pretty simple. Keyboard short cuts will also help to get some of the tasks done in a hurry.

I spoke with the author recently and he informed me that the program is available in english with english manuals. The program is sill waiting for a US distributor, but in the meantime you can purchase the program from: Jurgen Eckel, Sacklingerstr.2, D-40470 Dusseldorf, for about $15.

Before I leave you for another issue, here's a quick list of the available commands with a short explanation. If you're using geoShell, you might want to compare the commands with those CLI offers.

"backup"	archive command
"basic"	leaves GEOS and returns the computer to BASIC
"chkdsk"	check disk
"cls"	clear screen
"copy"	copy
"date"	date
"del"	delete files
"dir"	directory
"exit"	back to deskTop
"format"	format disk
"key"	F-key definition
"info"	GEOS Info
"label"	disk name
"pause"	for use with the batch editor
"print"	print
"ren"	file name
"restore"	restore a backup
"swap"	change drives
"sys"	saving CLI from memory to disk drive
"time"	reads the RTC
"type"	standard GEOS docs printed to screen
"ver"	CLIs version number
"xcopy"	contents of start disk copied
"xdel"	erase disk in default drive

HTML and the Web: Part II

By Katherine Nelson of Amiga Report Magazine

Welcome back to the next installment of our three-part series on HTML, the language of the World Wide Web. In the last issue, we explained a little about the Web, how to create a respectable homepage, and discussed some of the basic commands used in HTML. In this issue, we will continue with the tutorial by giving a more in-depth look at some of the HTML commands already mentioned, as well as a few new ones. In particular, we will begin looking at such advanced features as imagemaps, tables, and forms.

New Commands

A new version of the Web language called HTML3 is in development. While incorporating the commands in previous versions, it also adds some very powerful commands that increase the attractiveness of the pages they are used in. Currently, only some browsers such as Netscape for Mac/PC and IBrowse for the Amiga support these commands. However, they are becoming more and more common, and most graphical browsers will soon support them.

Since most of these advanced features rely on graphics and special text formatting, it is generally impossible to properly view such pages using text-based browswers like Lynx. Because of this, you should consider creating and linking some alternative text-specific pages for users with non-graphical browsers.

With that warning aside, let's look at a document with some of these commands in it. Note that our examples use a paragraph symbol (¶) to indicate where RETURN keypresses are located.

```
<HTML>¶
<HEAD>¶
<TITLE> Some Features of HTML3 </TITLE>¶
</HEAD>¶
<BODY BACKGROUND="back.gif" TEXT="#FFFFFF"
LINK="#FF0064" VLINK="#00FF64">¶
<H1>Some Features of HTML3</H1>¶
<IMG SRC="edge.gif" ALIGN="left">¶
<P ALIGN=CENTER> This paragraph should be center
ed. Also, the graphic above should be to the lef
t of all of this text. All the letters should be
white, except for links and visited links. For e
xample, here is a link to the menu of <A HREF="C
World.html">example pages</A> for this tutorial.
A link that you may not have visited yet would b
e <A HREF="http://www.efn.org/~kpw/cheesenet95/i
ndex.html">CheeseNet.</A><BR CLEAR="left">¶
<IMG SRC="edge.gif" ALIGN="right">¶
<P ALIGN="right">This paragraph should be right-
justified. Notice that this paragraph begins aft
er the left-edge picture ends. Also, the edge pi
cture is now on the right and is narrower.<BR CL
EAR="right">¶
<P ALIGN="left"> Another thing we can do is make
<FONT SIZE=+2>big text</FONT> or <FONT SIZE=-2>
small text </FONT>. This is useful for emphasizi
ng text.¶
</BODY>¶
</HTML>¶
```

What do all of these extra commands do? Rendered by an appropriate viewer, in this case Netscape, the page looks like Figure 1 at right.

Notice that instead of the usual gray background, there is an image tiled across the page. This is due to the BACKGROUND attribute in the <BODY> statement. Any picture can be used for this, but it is recommended that it be small for speed concerns. Background images should also be something which will contrast well with the text colors, so that the page will remain legible. All too often, Web authors tile unpleasant colors under their text, providing a blurry and disorienting effect.

If a solid color is all that is desired for a background, an easier and quicker way to accomplish this is to use BGCOLOR="*#rrggbb*" within the <BODY> tag. What this does is set the background color to the hexadecimal value *#rrggbb*, where *rr*, *gg* and *bb* are the hexadecimal codes for the red, green, and blue values respectively. For example, #FFFFFF would be white, and #A000FF would be a shade of purple (notice it mixes about 75% red [A0] with 100% blue [FF]).

The next attributes for the <BODY> tag follow the same pattern as the BGCOLOR attribute just mentioned. They each take the *#rrggbb* hexadecimal code as their values. The TEXT setting determines the color of the text on the page, LINK sets the color of the linked text on the page, and VLINK sets the color of a link already visited. These options are certainly not necessary to the development of an HTML document, but they really can make a page much more dynamic and exciting, as well as help to assure readability when using background colors or images.

The next new code is the option ALIGN="left" for the first picture. This sets the picture to the left of the text, as expected. Without this option, text would only be to the right of an image for one line, and the next line would just wrap around underneath the image. This option keeps as many lines of text as it can fit to the right of the image. With this option one can create decorative borders or whatever. Notice also that later in the example we use ALIGN="right" for a picture. This causes the picture to be to the right of all text, as expected.

Also notice that in places on the edge.gif picture the background shows through. This is done by creating the picture, and drawing with one color of your choice the areas you wish to be transparent. Then, use either your drawing program if it supports transparent GIFs, or a GIF utility to set that particular color transparent. Utilities that do this are available for many types of machines at various FTP sites. At the time of this writing, I was unable to find out if any of the popular 64 graphics manipulation packages would support this process.

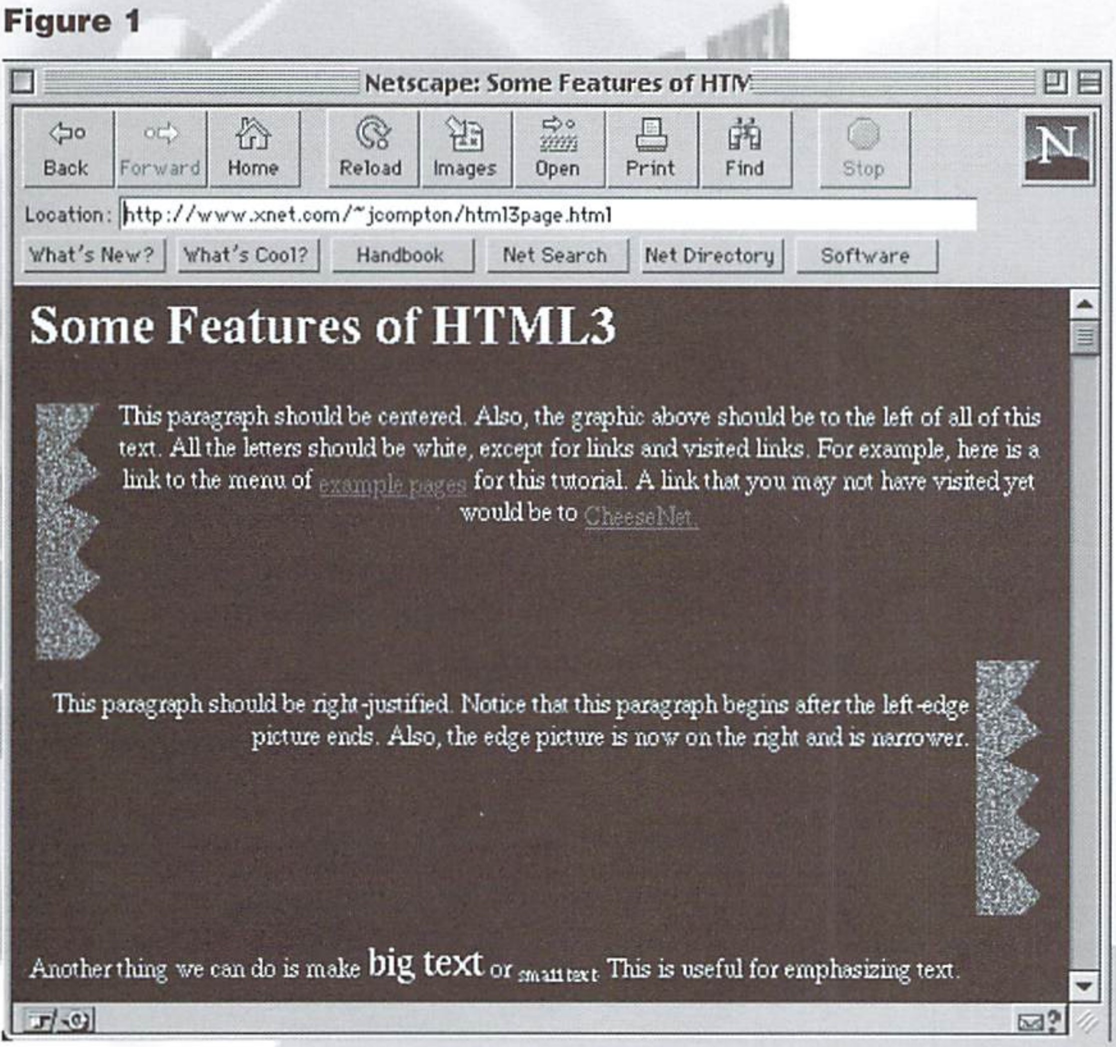
Following the first picture is the command <P ALIGN="center">. This command causes all the text in the paragraph to be centered horizontally on the page. The other allowed values for the ALIGN option in the <P> command include "left", "right", and "justify", the default being left.

A second way to center text is to use the <CENTER> and </CENTER> tags, with the text to be centered between them. This usage is becoming more widespread as more browsers support Netscape and other new commands.

The CLEAR attribute of the <BR> tag is particularly useful when left and right-aligning pictures. It simply signals the browser to begin the next line when one or more margins are clear, with possible settings being "left", "right", and "all", where the "all" option begins the next line at the point where both margins are clear. If these commands are not used, the browser will simply fit as much text next to an image as it can, which often is not the desired result. The same result can be obtained by using multiple <BR> or <P> tags, and guessing the amount of blank lines needed to get to the bottom of the picture, but this is not recommended, as people may set their viewer's window size to be larger or smaller than your own, which would foul up the formatting.

Finally, the <FONT> tag with the SIZE option provides a way to vary the size of the text you use in a document. This could be larger text for emphasis, or smaller text for copyright symbols and the like. Whatever reason you prefer, there are two ways to use this tag. One is to give absolute sizes, such as <FONT SIZE=7>, (7 is the largest size), or relative sizes, such as <FONT SIZE=+2>. This second example would set the text to be two sizes larger than the surrounding text. I happen to prefer the second method, but that's just a personal preference. Either way is perfectly fine, just don't forget to place a </FONT> tag after the text whose size you wish to change, or the rest of your document will all remain in that font size.

Figure 1



Location: http://www.xnet.com/~jcompton/html3page.html

Some Features of HTML3

This paragraph should be centered. Also, the graphic above should be to the left of all of this text. All the letters should be white, except for links and visited links. For example, here is a link to the menu of example pages for this tutorial. A link that you may not have visited yet would be to CheeseNet.

This paragraph should be right-justified. Notice that this paragraph begins after the left-edge picture ends. Also, the edge picture is now on the right and is narrower.

Another thing we can do is make big text or small text. This is useful for emphasizing text.

Imagemaps

Imagemaps, or "clickable images" are quickly becoming very popular. Instead of displaying a text menu of places to go, people would rather present a graphical menu, complete with fancy text and graphics. This is becoming increasingly popular on the Web's search reference sites, such as Yahoo and Lycos. Any GIF format picture may be used as an imagemap, even transparent GIFs. However, your Internet provider must be set up to handle imagemaps. Mine, for example, is not. So I do not have an example page for this particular HTML feature, but I will explain it.

You need to create a text file called a ".map file" (it must have the file extension ".map") It specifies geometric objects that divide the imagemap picture into regions. When you click on an area of the picture, the browser sends the mouse coordinates (relative to the upper-left corner of the picture) to the imagemap program. The program then checks these coordinates against the entries in the .map file to see if they are included in one of the objects. Here's an example of a .map file:

```
# Imagemap file for a picture¶
default http://www.xnet.com/~jcompton/CWorld.htm
l¶
rect http://www.xnet.com/~jcompton/rect.html 50,
30 70,80¶
circle http://www.xnet.com/~jcompton/circle.html
 20,40 5,40¶
poly http://www.xnet.com/~jcompton/poly.html 60,
10 40,45 10,45 80,20¶
point http://www.xnet.com/~jcompton/point1.html
75,90¶
point http://www.xnet.com/~jcompton/point2.html
3,3¶
```

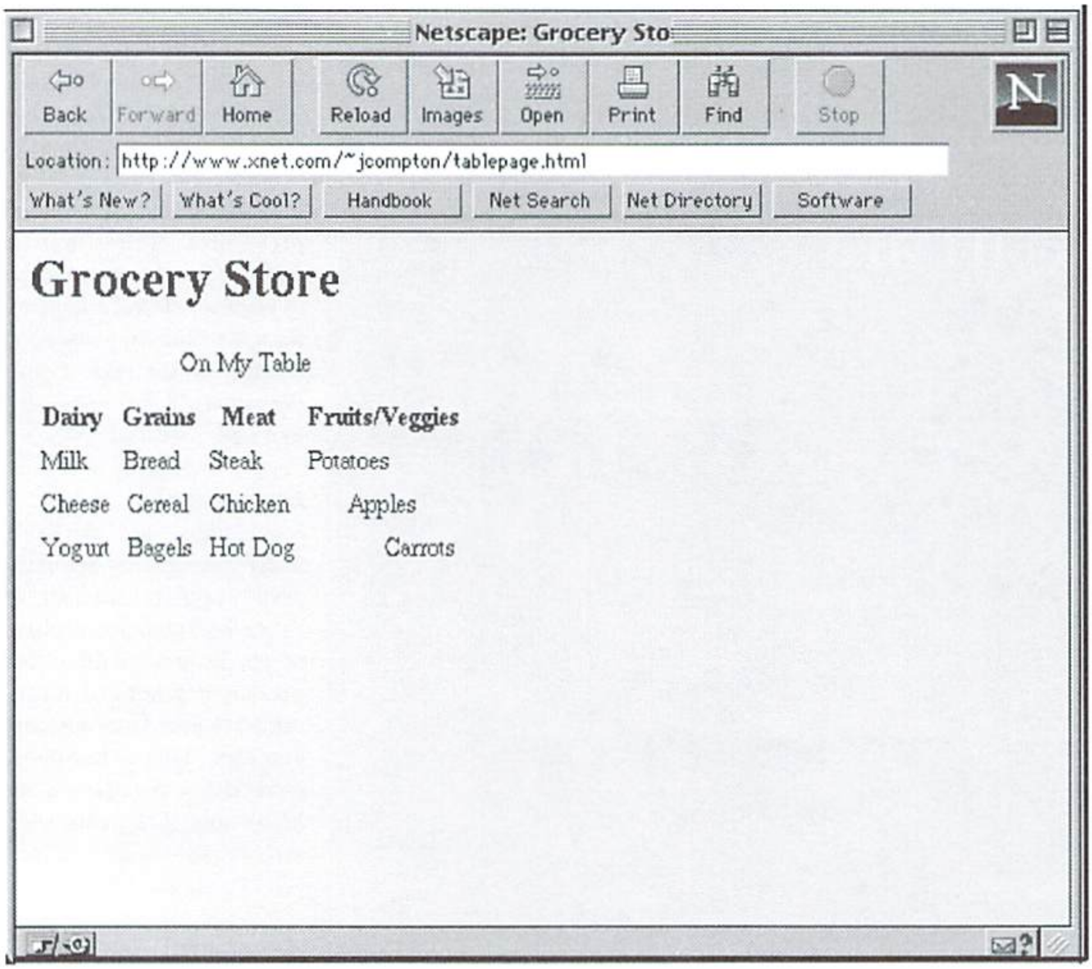
(The files listed in the example.map file, other than CWorld.html, are not real Web documents, just examples used for clarity.)

The first line in the example.map file is a comment, simply to let you know what the purpose of the file is. Comments in imagemap files must begin with the # character. The next line begins with the word "default", and it specifies the URL that the imagemap program should send you to if you don't click in one of the specified regions. In this particular file, this line is unnecessary, but that will be explained later. For the entries of the objects, the syntax is "object URL xcoordinate, ycoordinate". If there is more than one set of coordinates for an object, each set is separated by a space. "Object" refers to the object type, and "URL" is the destination address for that area of the image. Also note that objects are allowed to overlap. The first object that encompasses the mouse coordinates will be used, so be conscious of the order of the objects.

The first geometric object defined is a rectangle, with the object name "rect", and which requires two sets of coordinates. The first is the location of the upper-left hand corner, and the second is the location of the lower-right hand corner. Following the rectangle object is a circle object, with object name "circle", and also requires two sets of coordinates. The first set is the location of the center of the circle, and the second is the location of a point on the edge of the circle. The "poly" object is a polygon, with up to 100 coordinates that each reference a vertex of the polygon. The polygon will automatically "draw" the imaginary line between the first and last set of coordinates to close the object. (Remember, these objects will not actually appear on screen, but will be used as reference points to determine where the imagemap sends the user.)

The last of the object is the "point", which simply defines one point on the picture. If an area is clicked in that isn't defined, the imagemap will use the URL of the closest point object. For this reason, having only one point is silly, because you may as well just set a default location. This also explains why the default declaration is unnecessary in the .map file, because anywhere on the picture is considered to be close to one of the points, and the default URL will not be used.

Figure 2



http://www.xnet.com/~jcompton/tablepage.html

Grocery Store

On My Table

Dairy	Grains	Meat	Fruits/Veggies
Milk	Bread	Steak	Potatoes
Cheese	Cereal	Chicken	Apples
Yogurt	Bagels	Hot Dog	Carrots

Tables

There are two ways to show a table in HTML. The old way is to use the <PRE> and </PRE> tags. These tags surround text that will appear in the document in a fixed-width font exactly as typed, including spaces, carriage returns, or what have you. This is the only format that will show up with any sort of reliable results on a text-based browser such as Lynx. However, a new and more versatile way is to use the syntax that is actually called a table. An HTML document, which takes advantage of some of the commands we discussed earlier, with a table is presented here:

```
<HTML>¶
<HEAD>¶
<TITLE>Grocery Store</TITLE>¶
</HEAD> ¶
<BODY BGCOLOR="#EFEFEF" TEXT="#000050" LINK=
"#009020" VLINK="000080">¶
<H1>Grocery Store</H1>¶
<TABLE>¶
<CAPTION ALIGN = "center" > On My Table </CA
PTION>¶
<TR> <TH>Dairy</TH> <TH>Grains</TH> <TH>Meat
 </TH> <TH>Fruits/Veggies</TH> </TR>¶
<TR ALIGN="left" VALIGN="top"> <TD>Milk</TD>
 <TD>Bread</TD> <TD>Steak</TD> <TD>Potatoes<
/TD> </TR>¶
<TR ALIGN="center" VALIGN="middle"> <TD>Chee
se</TD> <TD>Cereal</TD> <TD>Chicken</TD> <TD
>Apples</TD> </TR>¶
<TR ALIGN="right" VALIGN="bottom"> <TD>Yogur
t</TD> <TD>Bagels</TD> <TD>Hot Dog</TD> <TD>
Carrots</TD> </TR>¶
</TABLE>¶
</BODY>¶
</HTML>¶
```

This document is rendered in a browser as show in Figure 2 at left.

A table must be enclosed by the <TABLE> and </TABLE> elements. The <TABLE> tag can also include the one-word attribute BORDER, which in HTML3 places a dividing line between the different cells of the table. The first tag used is the <CAPTION> </CAPTION> set. This, as you might expect, will determine the caption to be used for the table.

The next tag is <TR> with its companion </TR>. These define a row of the table. They can include ALIGN and VALIGN options, which determine the alignment of text in the cells in the horizontal and vertical directions, respectively. The ALIGN attribute can take the values "left" "right" and "center", while the VALIGN attribute uses the values "top" "bottom" and "middle". The other tags used in a table are <TH> and <TD>. These tags do not require the closing tags, but they are used with them above because it is considered good form. The <TH> tag delineates a header for the table. The <TD> tag separates the data in the table. Both of these tags must be included within a <TR> tag. Also, these tags can use the ALIGN and VALIGN options described above, and the options in <TH> and <TD> override those in the <TR> tag that contains them.

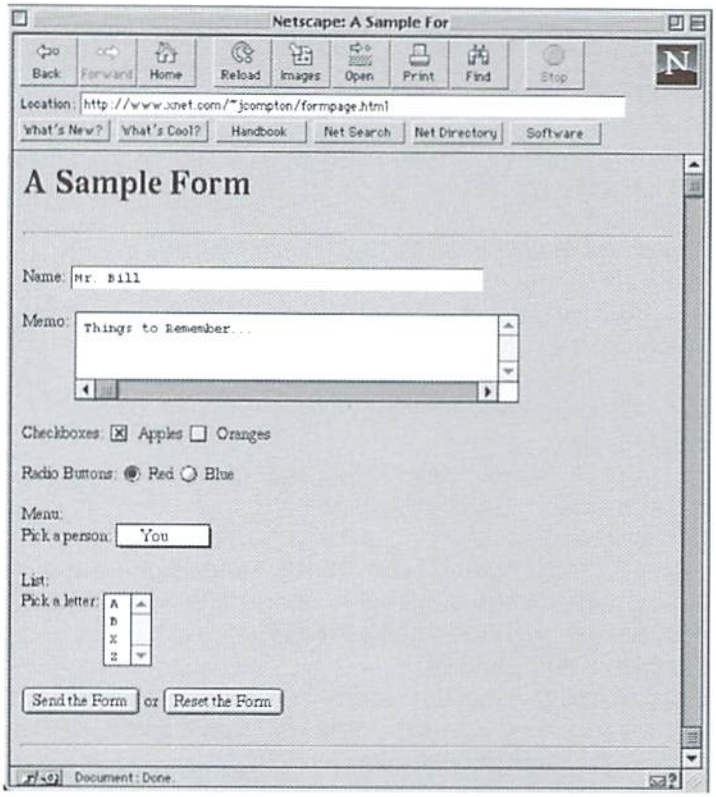
Forms

Forms are fairly widely supported at this point, mainly for using search engines or sending feedback to the owner of a web page. However, in order for you to be able to incorporate forms in your own page, your server must support this feature. Again, mine does not, but here I have used a different server that sends the values inputted into the form back to the viewer. This doesn't accomplish much, but it demonstrates how a basic form works. Thankfully, the art of form-ing has not evolved beyond the simple bounds that Lynx can support, so such pages are available to virtually all Web browsers.

```
<HTML>¶
<HEAD>¶
<TITLE>A Sample Form</TITLE>¶
</HEAD>¶
<BODY>¶
<H1>A Sample Form</H1>¶
<P><HR>¶
<FORM METHOD="POST"¶
ACTION="http://hoohoo.ncsa.uiuc.edu/htbin-po
st/post-query">¶
<P>Name:¶
<INPUT TYPE="text" NAME="Name" VALUE="Mr. Bi
ll" SIZE=50 MAXLENGTH=75>¶
<P>Memo:¶
<TEXTAREA NAME="Memo" ROWS=3 COLS=50>Things
to Remember...</TEXTAREA>¶
<P>Checkboxes:¶
<INPUT TYPE="checkbox" NAME="Apples" VALUE="
Tasty" CHECKED> Apples <INPUT TYPE="checkbox
" NAME="Oranges" VALUE="Tasty"> Oranges¶
<P>Radio Buttons:¶
<INPUT TYPE="radio" NAME="Color" VALUE="Red"
 CHECKED> Red <INPUT TYPE="radio" NAME="Colo
r" VALUE="Blue"> Blue¶
<P>Menu:¶
<BR>Pick a person:¶
<SELECT NAME="Person">¶
<OPTION> You <OPTION> Me¶
<OPTION> He <OPTION> She¶
</SELECT>¶
<P>List:¶
<BR>Pick a letter:¶
<SELECT NAME="Letter" SIZE=4 MULTIPLE>¶
<OPTION> A <OPTION> B¶
<OPTION> X <OPTION> Z¶
</SELECT>¶
<P>¶
<INPUT TYPE="submit" VALUE="Send the Form">
or <INPUT TYPE="reset" VALUE="Reset the Form
">¶
</FORM>¶
<HR>¶
</BODY>¶
</HTML>¶
```

This form appears as shown in Figure 3 on the next page.

The form begins with the tag <FORM>. This tag takes two attributes. The first is METHOD. This can be set to GET or POST. The differences will be explained in the next article in this series, which concerns slightly more advanced subjects. It also uses the attribute ACTION. This is set to the URL of the program or script that processes the form information. You can use a URL specified by your server, which can serve different functions depending on the server. Most prefer to automatically send you the form results in e-mail. If your server does not yet support forms, you can use the URL that I have to test your form abilities.



Location: http://www.xnet.com/~jcompton/formpage.html

A Sample Form

Name: Mr. Bill

Memo: Things to Remember...

Checkboxes: ☒ Apples □ Oranges

Radio Buttons: ● Red ○ Blue

Menu:
Pick a person: You

List:
Pick a letter: A B X Z

[Send the Form] or [Reset the Form]

Figure 3

The tag used most often in this document is the <INPUT> tag. There are six different types of input, "text", "password", "checkbox", "radio", "submit", and "reset". The text and the password options are identical except the characters typed in a password field are shown with asterisks or a similar character to protect secrecy. All these types except for "submit" and "reset" take a NAME option. This is the variable name for the data entered. Notice that the two radio types have the same NAME. This causes a "one-of-many" effect, allowing the single variable given in NAME to have only one of the given values. A second option is VALUE. In the text and password types, this causes default text to appear. In the checkbox and radio types, VALUE is the value of the variable when it is checked. With the submit and reset types, VALUE sets the label of the button. The checkbox and radio types can also take a CHECKED option. If this option is present, it causes the item to be selected. The last two options present, SIZE and MAXLENGTH, only apply to the text and password fields. They set the maximum number of characters in the input area and the maximum characters allowed, respectively.

The second type of tag used in a form is the <TEXTAREA> tag. A <TEXTAREA> is a text entry area with no limit to the number of characters. It requires a NAME option similar to above. It also requires ROWS and COLS options. These options are set to the number of characters down and across you wish to display at once. Also, within a <TEXTAREA>, all spacing and carriage returns are respected. Any default text should be placed between the <TEXTAREA> and </TEXTAREA> tags. If no default text is desired, put nothing between them. This type of form entry is useful for writing comments and feedback.

The next type of selection tag is <SELECT>. This creates either a menu, such as in the first example, or a list, such as in the second example. The only difference between them is the SIZE tag. If SIZE is greater than 1, a list is generated. If not, a menu is used. If the MULTIPLE option is included, which allows for selecting of more than one entry, a list is automatically used. The NAME option is used in this tag the same way it is used in a checkbox INPUT type. Between the <SELECT> and </SELECT> are a list of <OPTION>s. The text following each <OPTION> tag is a separate choice. The <OPTION> tag can also include an attribute of SELECTED to give a default value.

Try not to feel daunted if the amount of commands in HTML seems a little overwhelming. The hallmark of a good Web page is not so much the gimmicky layout as it is the information presented. Nevertheless, good presentation won't hurt, either. In the next issue's concluding article, we'll examine the design of more complex forms and touch on CGI scripts, perhaps the most intricate and powerful aspect of the Web.Until then, enjoy creating those Web pages!

Check Out These Web Sites for More Information about HTML:

WWW Viewer Test Page
http://www-dsed.llnl.gov/documents/
Harbinger Net Services (HTML Evaluation)
http://www.harbinger.net/html-val-svc/
HTML pointers
http://netspace.cis.brown.edu/netspace/wwwdoc.html
HTML things
http://www.csv.warwick.ac.uk/~stuce/tech.html
HTML Documentation Table of Contents
http://www.utirc.utoronto.ca/HTMLdocs/NewHTML/htmlindex.html
Fill out forms example
http://www.wfbr.edu:8002/www/testform.html
Homepage Creation!
http://the-inter.net/www/future21/html.html
HTML+ Discussion Document
http://info.cern.ch/hypertext/WWW/MarkUp/HTMLPlus/htmlplus_1.html
The World Wide Web Initiative: The Project
http://www.w3.org/hypertext/WWW/

Find out more about Lynx from the following sites:

A crude LYNX tutorial
http://www.gallaudet.edu/~kjcole/tutorial.html
Lynx-Dev Discussion List
http://access5.digex.net/~asgilman/lynx-dev/about_lynx-dev.html
Lynx-Dev Archive Index
http://lynx.cc.ukans.edu/lynx-dev/
The Lynx Enhanced Pages Index
http://www.nyu.edu/pages/wsn/subir/lynx.html
Lynx, The
http://www.gold.net/lynx/

- GRM

... And HTML for All!

Catering to Lynx and other text-based WWW Browsers

When designing a web site, seriously consider who will be accessing your site, and the type of Browser they may be using. If you are setting up a site supporting Commodore users, you will want to be able to give all users the ability to experience your site, no matter what type of browser they use. There are times when using the latest and greatest HTML commands will leave those you are trying to impress disenchanted and left in the cold. For instance, if you're designing a sales-oriented web site, you'd want to offer your products (and images of your product) to anyone that visits your site. To avoid alienating your visitors try the following suggestions.

Graphic Images

There are three methods of adding graphics to your WWW site that will offer something meaningful to text based Browsers.

You can display the image, and offer a separate link to the image so it may be downloaded.

Example:

```
<img src="image.gif"></a>
<a=href"image.gif">image.gif</a>
```

Another option many use is to tell text users what they're missing by adding text to be displayed. Instead of the usual [IMAGE], the example given below would display [Photo] when viewed by text browsers.

```
<IMG SRC="image.gif" ALT="[Photo]"> </A>
```

This is like showing someone a treat but not letting them have it. Okay, so your logo is probably no big deal, but those who can't see it don't know that. With a little added text you can offer the best of both worlds by having the image shown and a selectable link together in the same anchor:

```
<A HREF="image.gif"><IMG SRC="image.gif"
ALT="[Photo of XYZ]"> </A>
```

This shows the graphic, and any visitors with a text browser can download it for later viewing. The brackets aren't needed, but they give a clue that it's an image link.

You can include formatting commands too, as they only affect how the image is shown. It doesn't change the image that's downloaded. For Commodore users designing a web page, formatting the graphic can be a challenge, as you won't be able to see the effects of word wrap around the graphic. However, because graphic browsers use different fonts of varying sizes, there's not much point in worrying too much over how the text wraps. To format an image so it's about the size of a wallet photograph and displayed on the left side of the screen, the above example can be expanded to:

```
<A HREF="image.gif"><IMG SRC="image.gif"
ALT="Photo" HSPACE="20" VSPACE="20"
ALIGN="LEFT"> </A>
```

Clear Navigation for All

If your web site uses graphics for menu options, the above examples can help text users navigate through your site. Replace <A HREF="image.gif"> with <A HREF="thispage.html"> and change the alternate text to something appropriate for the link. Graphic users will see the images you've selected for each menu option, and text users will have meaningful text to select from instead of a jumble of [IMAGE] icons on the screen. With a few additional characters, your site can be transformed from something that appears meaningless when viewed by a text browser:

```
[IMAGE] [IMAGE]
[IMAGE] [IMAGE]
```

Into something much more meaningful:

```
[Joe's Garage]    [Specials]
[Customer Service] [Mechanics Forum]
```

Formatted Text

An easy way to display a data table is to use the <PRE> command. The <PRE> is placed before the formatted text, and </PRE> after it. The table of data is shown as you've formatted it, without any additional commands, and is accessible to everyone.

Example:

```
<H1>Grocery Store</H1>
<PRE>            On My Table

Dairy    Grains   Meat      Fruits/Veggies
Milk     Bread    Steak     Potatoes
Cheese   Cereal   Chicken   Apples
Yogurt   Bagels   Hot Dog   Carrots
</PRE>
```

Notes: (1) There are 12 spaces between the <PRE> tag and the "On My Table" text. (2) Spaces are used to align the text. In this example, all columns are left justified. (3) Any hard returns located between the <PRE> and </PRE> tags will be taken literally (HTML normally ignores hard returns).

- Gaelyne R. Moranec

Interlace? No Worries!

Have Interlaced GIFS got you down? Here's a quick cure for C128 users! Download GIFDELACE.SFX (on some systems it's named DELACER.SFX) and use it to convert that pesky interlace into a standard GIF file.

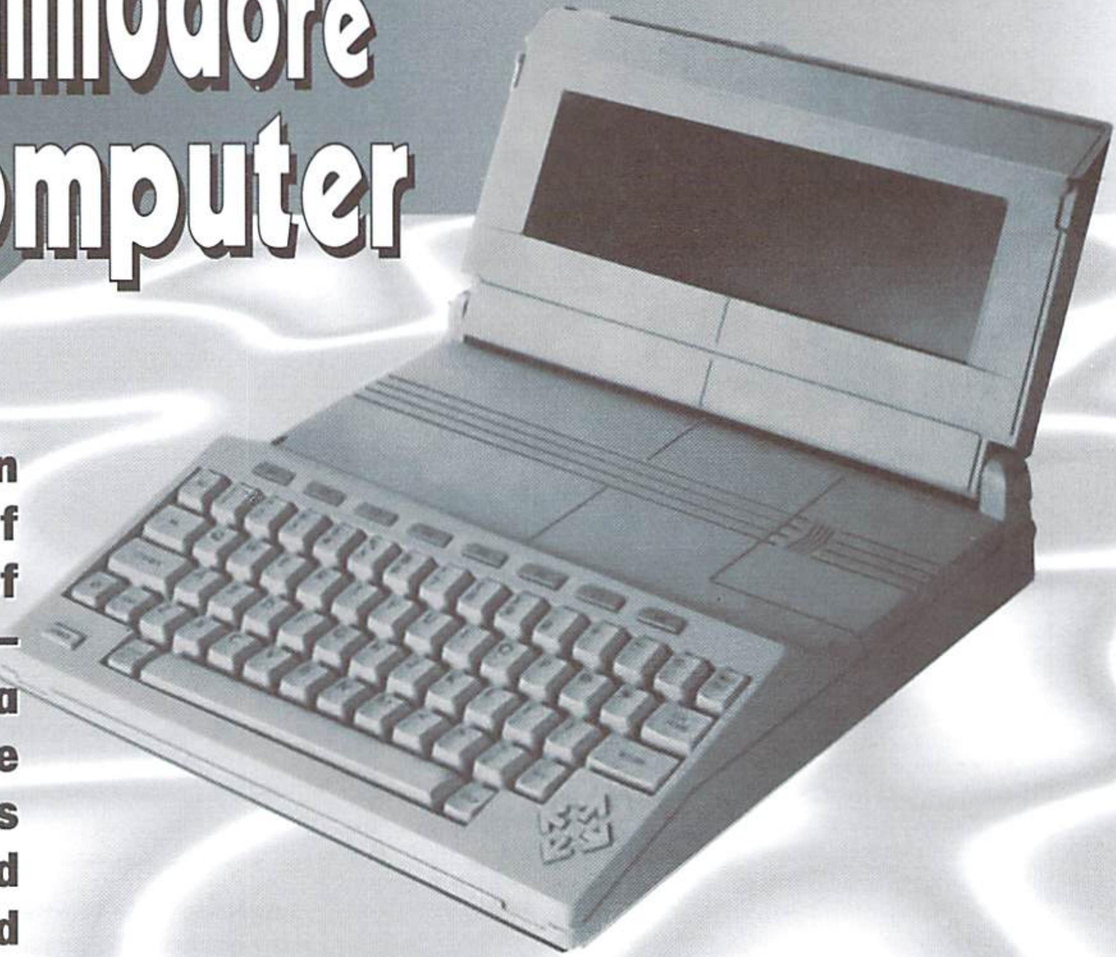
- GRM

HARDWARE Collectibles

By Jim Brain

The Commodore LCD Computer

This rare orphan sits on a shelf somewhere out of the public's eye– a reminder of a time when people expected less from laptops and they delivered more...



Does anyone remember the Commodore SX-64? It was one of the first Commodore "portable" computers. It weighed quite a bit, as it contained an entire Commodore 64, a 1541 disk drive, a 5" color monitor, and the associated power supplies, cables and case all in one unit. People called it and the computer like it "luggables", since that is how they were carried. Also, the SX-64 was not really portable, since it required AC line voltage to operate. So much for using your 64 on the road!

In the mid 1980's, Commodore again tried their hand at portable computing. This time, CBM noted that to be truly portable, a computer must be lightweight and able to run on internal power. The engineers set forth to deliver a prototype, and the Commodore LCD (CLCD) Laptop computer was born.

This technical marvel borrowed much of its styling, features, and functionality from the Commodore Plus/4 (264 series) line of computers. Yet, it had a personality all its own. The unit sported the following features:

- A sleek, beige case styled similar to the Plus/4
- Plus/4 style-72 key keyboard layout, complete with real keys (not chicklet), 8 function keys and 4 cursor keys
- An 80 character by 16 line LCD screen that flips down when not in use
- 8 internal integrated software programs
- An industry standard IEA-232 (RS-232) port
- An industry standard Centronics parallel (printer) port
- An HP bar code reader port
- 96 kilobytes of ROM
- 32 kilobytes of battery backed RAM
- A super set of Plus/4 BASIC 3.5 called BASIC 3.6
- A 65C102 CPU (based on the Western Design Center 65C02 CPU core under license to CBM)
- An expansion port
- A Commodore serial port
- An internal 300 bps auto dial/auto answer modem.
- Able to run on 4 AA batteries.
- Is 2.18" tall by 11" deep by 11.75" wide, and weighs 5 pounds

Unlike its predecessor, the SX-64, the CLCD was not designed to compete with the Commodore 64—or be compatible with it. Different from the 64, it can only utilize some serial bus peripherals, some BASIC programs, and some data files. Since the model is not forced to be compatible, it can sport new features like the serial and parallel ports and the larger LCD screen.

The LCD screen was somewhat plain by itself, but its presence removed the need to carry or find a suitable monitor to display information. The 80x16 screen could be used as a 61440 pixel bitmap, or could be used in text modes to display 1200 characters of information. The user of this machine could organize all that information quickly, thanks to the built in software suite of programs. There were eight programs available:

- Word Processor
- File Manager
- Spreadsheet
- Address Book
- Scheduler
- Calculator
- Memo Pad
- Telecommunications Program

All were somewhat sparse in features, much like the built-in offerings for the Plus/4, but would function nonetheless. Of special notice was the Spreadsheet, similar in function to Multiplan, the terminal program, which offered VT-52 emulation, and the Calculator and Memo Pad programs, which could be used from within any application. The battery backed RAM allowed the user to compute all day without saving files to disk or hassling with other forms or magnetic media.

To the seasoned Commodore user, little would look out of place. The Plus/4 user would feel at ease with the keyboard layout, the BASIC programmer would see all the regular BASIC 2.0 commands plus the 3.5 additions, including the BASIC 4.0 disk commands. The unit still featured the Commodore editing environment, and machine language programmers would notice all the old 6502 opcodes were still present, although 8 new instruction and some new addressing modes were also available. The end user would enjoy the addition of standard RS-232 and parallel ports, and the existing Commodore user will find that his or her existing Commodore peripherals can be used with the new machine.

About the only thing missing from the unit was full compatibility with any of Commodore's previous models. Still, the unit was ideally suited to the utilitarian computer environment, where such issues may not exist. Another thing bothering Commodore enthusiasts was the name. Some people were calling it the Commodore 64LCD or the C128LCD laptop computer. Neither was really correct, since this unit shares the most compatibility with the Plus/4 machines. Commodore is partly to blame for this, as they leaked reports of a 64 laptop early in the development cycle.

To aid in securing a part of the utilitarian market, the unit's HP bar code reader port made the unit desirable for shop and warehouse managers, who needed to move stock and constantly update their inventory. Other users included business travelers, secretaries, and executives, as well as other people who were on-the-go and needed computing power wherever they were.

Although the unit did not come with a means to store files and data, a user could utilize any of the Commodore disk drives to off-load data. Also, the expansion port on the unit could accommodate extra RAM and/or ROM cards. If remote access or connection was desired, the built in 300 bps modem could be used with any telephone line to connect to remote locations and systems. The VT-52 terminal emulation allowed a wider variety of programs to be run over the modem, and the 300 bps speed was speedy for the era.

About the only thing wrong with this unit was its timing. At the time of introduction, the Commodore 64 had firmly rooted itself as the only Commodore computer. The Plus/4 and C16 suffered from this same viewpoint, as business users stayed away from CBM computers, thinking they were only "toys". So, the Commodore LCD wasted away into nothingness. Only one unconfirmed sighting has been reported, and I suspect that unit is the only prototype ever manufactured. It's a shame the unit never caught on, because the weight and capabilities of the unit would come as welcome additions to the Commodore users. Also, the use of the WDC 65C02 core allowed more flexibility in writing software.

So, this rare orphan sits on a shelf somewhere, out of the public's eye. A reminder of a day long past, when people expected less from laptops and they delivered more.

√Σ The program listing in this article can be verified with our new CheckSum utility. See page 54.

Intelligence IQ Quest

by Doug Cotton

Program Requirements
C-64 or C-128 in 64 mode, mouse in port 1, or joystick in port 2.

Okay, so it's no intergalactic mission, there are no mutant zombies, and the Earth isn't doomed to certain destruction unless you can find some way to stop it. Intelligence Quest is, quite simply, one of those annoying little puzzle games that challenges you to do the seemingly impossible.

Most of you have probably played similar games, usually found in the form of a little piece of wood with holes drilled in it, which you fill with pegs. Leaving one hole blank at the start, you attempt to jump one peg over another—much like jumping in the game Checkers—until there are no adjacent pegs left. The fewer pegs you have left at this point, the better your score. Take this concept, change the board to a computer screen, change the pegs to little balls and the holes to square cells, and there you have it.

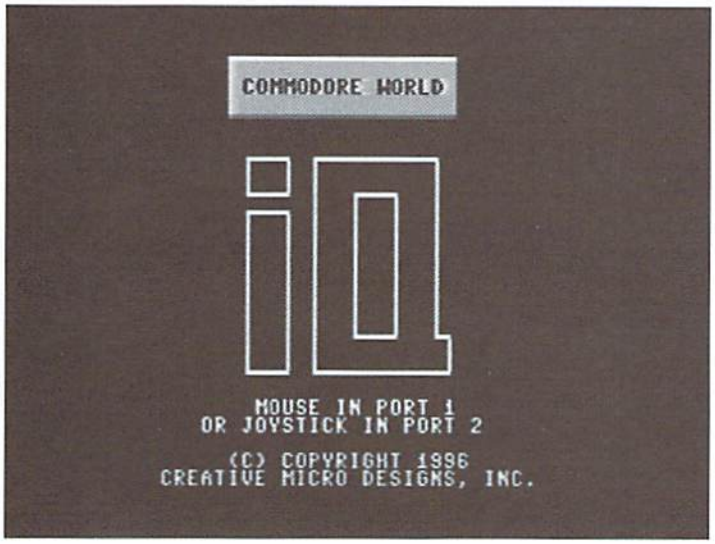
Playing The Game

Once you have the program typed in, verified and saved, you're ready to start it up. The program requires either a mouse (in port 1) or joystick (in port 2). If you plan to use a mouse, make sure it is plugged into port 1 before you start the program—if you don't, it won't be recognized.

After the title screen clears, you're ready for the first turn. The game will leave one (random) cell open. Each turn consists of using the pointer to make two selections. First, select the ball you wish to move by positioning the tip of the pointer over it, then clicking once on the fire button (or left mouse button). Next, select the open cell you want to move the selected ball to. If the move is legal (the selected ball must jump over another ball to end up in the empty cell), the ball will be moved and you move on to the next turn.

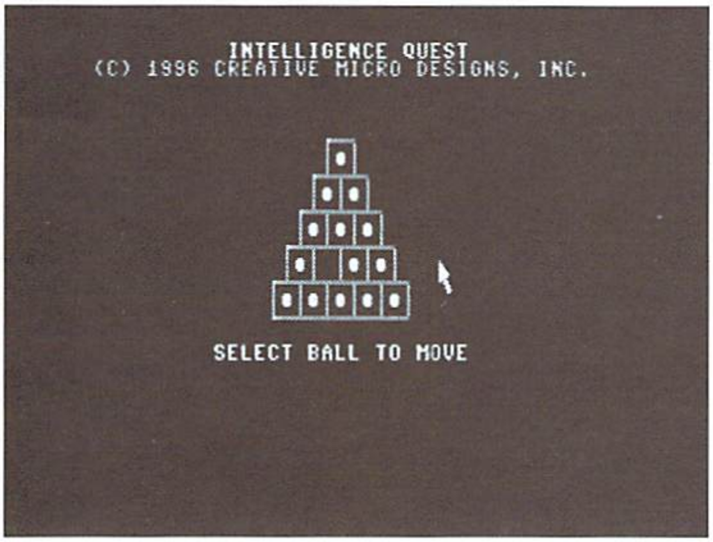
Scoring

When there are no legal moves left, the program will tally up the number of balls left remaining and evaluate your performance. Leave only one ball, or eight balls with no moves left) and you're a certified genius!

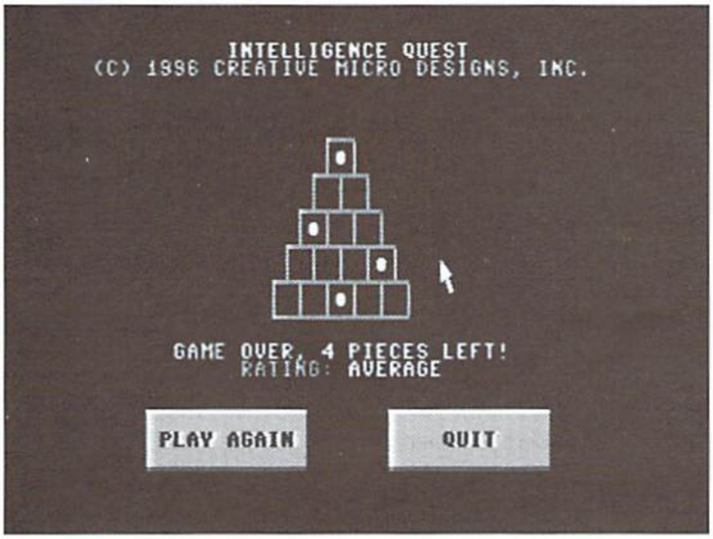


MOUSE IN PORT 1
OR JOYSTICK IN PORT 2

(C) COPYRIGHT 1996
CREATIVE MICRO DESIGNS, INC.


INTELLIGENCE QUEST
(C) 1996 CREATIVE MICRO DESIGNS, INC.

SELECT BALL TO MOVE


INTELLIGENCE QUEST
(C) 1996 CREATIVE MICRO DESIGNS, INC.

GAME OVER, 4 PIECES LEFT!
RATING: AVERAGE

```
√Σ   IQ.BAS

159  100 rem mini-iq (c) 1996 by cmd
168  110 :
232  120 poke53280,0:poke53281,0
26   130 print "{CLEAR/HOME}{WHT}{11 SPACES}{
     17 CMDR p}"
164  140 print "{LGRY}{11 SPACES}{RVS ON}{17
     SPACES}{RVS OFF}"
20   150 print "{11 SPACES}{RVS ON} commodore
      world {RVS OFF}"
175  160 print "{11 SPACES}{RVS ON}{17 SPACES
     }{RVS OFF}"
70   170 print "{DGRY}{11 SPACES}{17 CMDR y}"
162  180 print"{CRSR DN}{WHT}{12 SPACES}{CMDR
      a}{2 SHFT *}{CMDR s} {CMDR a}{8 SHFT *}
     {CMDR s}"
89   190 print"{12 SPACES}{SHFT -}{2 SPACES}{
     SHFT -} {SHFT -}{8 SPACES}{SHFT -}"
118  200 print"{12 SPACES}{CMDR z}{2 SHFT *}{
     CMDR x} {SHFT -}{2 SPACES}{CMDR a}{2 SHF
     T *}{CMDR s}{2 SPACES}{SHFT -}"
186  210 print"{12 SPACES}{CMDR a}{2 SHFT *}{
     CMDR s} {SHFT -}{2 SPACES}{SHFT -}{2 SPA
     CES}{SHFT -}{2 SPACES}{SHFT -}"
240  220 print"{12 SPACES}{SHFT -}{2 SPACES}{
     SHFT -} {SHFT -}{2 SPACES}{SHFT -}{2 SPA
     CES}{SHFT -}{2 SPACES}{SHFT -}"
234  230 print"{12 SPACES}{SHFT -}{2 SPACES}{
     SHFT -} {SHFT -}{2 SPACES}{SHFT -}{2 SPA
     CES}{SHFT -}{2 SPACES}{SHFT -}"
28   240 print"{12 SPACES}{SHFT -}{2 SPACES}{
     SHFT -} {SHFT -}{2 SPACES}{SHFT -}{2 SPA
     CES}{SHFT -}{2 SPACES}{SHFT -}"
22   250 print"{12 SPACES}{SHFT -}{2 SPACES}{
     SHFT -} {SHFT -}{2 SPACES}{SHFT -}{2 SPA
     CES}{SHFT -}{2 SPACES}{SHFT -}"
8    260 print"{12 SPACES}{SHFT -}{2 SPACES}{
     SHFT -} {SHFT -}{2 SPACES}{SHFT -}{2 SPA
     CES}{SHFT -}{2 SPACES}{SHFT -}"
2    270 print"{12 SPACES}{SHFT -}{2 SPACES}{
     SHFT -} {SHFT -}{2 SPACES}{SHFT -}{2 SPA
     CES}{SHFT -}{2 SPACES}{SHFT -}"
12   280 print"{12 SPACES}{SHFT -}{2 SPACES}{
     SHFT -} {SHFT -}{2 SPACES}{CMDR z}{2 SHF
     T *}{CMDR x}{2 SPACES}{CMDR z}{CMDR s}"
89   290 print"{12 SPACES}{SHFT -}{2 SPACES}{
     SHFT -} {SHFT -}{9 SPACES}{SHFT -}"
207  300 print"{12 SPACES}{CMDR z}{2 SHFT *}{
     CMDR x} {CMDR z}{9 SHFT *}{CMDR x}"
88   310 print"{CRSR DN}{13 SPACES}{YEL}mouse
      in port 1"
157  320 print"{9 SPACES}or joystick in port
     2 "
125  330 print"{CRSR DN}{LGRY}{11 SPACES}(c)
     copyright 1996"
85   340 print"{6 SPACES}{CYN}creative micro
     designs, inc.{HOME}";
153  350 :
77   360 rem setup
173  370 :
139  380 dimm(15,15),p(15),mv(9,9)
176  390 sx=53248:sz=53264:sy=53249:me=49152:
     rt$="{6 CRSR RT}"
101  400 ra$(5)="below average":ra$(4)="avera
     ge":ra$(3)="above average"
193  410 ra$(2)="bright":ra$(1)="genius":md$=
     "{HOME}{17 CRSR DN}"
176  420 bl$="{39 SPACES}"
10   430 gosub1680
243  440 :
162  450 fori=1to15:forj=1to15:readm(i,j):nex
     t:next
```

√Σ IQ.BAS *(cont.)*

```
4      460 :
62     470 fori=1to9:forj=1to9:readmv(i,j):next
       :next
169    480 l=49152
201    490 readd:ifd=-1then510
23     500 pokel,d:l=l+1:goto490
177    510 sys(me)
64     520 :
97     530 print"{CLEAR/HOME}{WHT}{11 SPACES}in
       telligence quest"
197    540 print"{CYN} (c) 1996 creative micro
       designs, inc."
117    550 fori=1to15:p(i)=209:next
156    560 r=int(rnd(0)*15)+1:p(r)=32:n=14
118    570 :
3      580 rem main loop
138    590 :
157    600 gosub1320
166    610 gosub1230
6      620 print
178    630 :
189    640 rem input values
198    650 :
77     660 printmd$;"{WHT}{10 SPACES}select bal
       l to move"
226    670 gosub1610:sp=mv
224    680 :
135    690 printmd$;bl$;md$;"{WHT}{8 SPACES}sel
       ect square to move to"
22     700 gosub1610:lz=mv:printmd$;bl$
3      710 :
38     720 rem test values
23     730 :
255    740 ifsp<1orsp>15orlz<1orlz>15then980
227    750 if(p(lz)<>32)or(p(sp)<>209)then980
53     760 rp=0:fori=1to15:ifm(sp,i)=lzthenrp=i
135    770 next
221    780 ifrp=0then980
214    790 ifp(rp)=32then980
89     800 :
57     810 rem do turn
109    820 :
203    830 p(rp)=32:p(sp)=32:p(lz)=209:n=n-1
129    840 :
61     850 rem check for more moves
149    860 :
166    870 i=0
175    880 i=i+1:ifi=16then1040
252    890 ifp(i)=32then880
199    900 j=0
22     910 j=j+1:ifj=16then880
26     920 ifm(i,j)=0then910
19     930 if(p(j)=209)and(p(m(i,j))=32)then610
255    940 goto910
243    950 :
60     960 rem messages
0      970 :
118    980 printmd$;"{13 SPACES}{LRED}illegal m
       ove!";
157    990 gosub1940:fori=1to500:next
202    1000 printmd$;bl$:goto610
40     1010 :
182    1020 rem game over
68     1030 :
10     1040 gosub1230:printmd$;"{YEL}{7 SPACES}
       game over,{WHT}";n;"{YEL}pieces left!"
167    1050 ifn=8thenn=1
216    1060 ifn>4thenn=5
112    1070 ra$="{LRED}rating: {WHT}"+ra$(n):rl
       =len(ra$)-2:rs=20-int(rl/2)-1
8      1080 printspc(rs);ra$
```

√Σ IQ.BAS *(cont.)*

```
55     1090 printmd$;"{3 CRSR DN}";
107    1100 print"{5 SPACES}{WHT}{12 CMDR p}{6
       SPACES}{12 CMDR p}"
52     1110 print"{5 SPACES}{LGRY}{RVS ON}{12 S
       PACES}{RVS OFF}{6 SPACES}{RVS ON}{12 SPA
       CES}{RVS OFF}"
179    1120 print"{5 SPACES}{RVS ON} play again
        {RVS OFF}{6 SPACES}{RVS ON}{4 SPACES}qu
       it{4 SPACES}{RVS OFF}"
48     1130 print"{5 SPACES}{RVS ON}{12 SPACES}
       {RVS OFF}{6 SPACES}{RVS ON}{12 SPACES}{R
       VS OFF}"
94     1140 print"{5 SPACES}{DGRY}{12 CMDR y}{6
        SPACES}{12 CMDR y}{HOME}";
188    1150 :
204    1160 gosub1500:ifr<22orr=25then1160
174    1170 ifc>5andc<18then530
188    1180 ifc>23andc<36thenprint"{CLEAR/HOME}
       {WHT}":sys(me):poke198,0:new
235    1190 goto1160
238    1200 :
105    1210 rem piece display subroutine
250    1220 :
104    1230 print"{HOME}{6 CRSR DN}{YEL}";:i=1
23     1240 forl=1to5:print"{15 CRSR RT}";left$
       (rt$,5-l);
178    1250 forc=1tol
251    1260 printchr$(p(i));"{CRSR RT}";
21     1270 i=i+1:next
223    1280 print:print
210    1293 next:print"{WHT}";
163    1300 gosub1760:return
93     1310 :
146    1320 rem board display subroutine
105    1330 :
54     1340 print"{HOME}{5 CRSR DN}{GRN}";
148    1350 print"{18 SPACES}{CMDR a}{SHFT *}{C
       MDR s}"
178    1360 print"{18 SPACES}{SHFT -} {SHFT -}"
221    1370 print"{17 SPACES}{CMDR a}{CMDR e}{C
       MDR r}{CMDR e}{CMDR s}"
50     1380 print"{17 SPACES}{SHFT -} {SHFT -}
       {SHFT -}"
142    1390 print"{16 SPACES}{CMDR a}{CMDR e}{C
       MDR r}{CMDR e}{CMDR r}{CMDR e}{CMDR s}"
114    1400 print"{16 SPACES}{SHFT -} {SHFT -}
       {SHFT -} {SHFT -}"
216    1410 print"{15 SPACES}{CMDR a}{CMDR e}{C
       MDR r}{CMDR e}{CMDR r}{CMDR e}{CMDR r}{C
       MDR e}{CMDR s}"
51     1420 print"{15 SPACES}{SHFT -} {SHFT -}
       {SHFT -} {SHFT -} {SHFT -}"
209    1430 print"{14 SPACES}{CMDR a}{CMDR e}{C
       MDR r}{CMDR e}{CMDR r}{CMDR e}{CMDR r}{C
       MDR e}{CMDR r}{CMDR e}{CMDR s}"
19     1440 print"{14 SPACES}{SHFT -} {SHFT -}
       {SHFT -} {SHFT -} {SHFT -} {SHFT -}"
106    1450 print"{14 SPACES}{CMDR z}{SHFT *}{C
       MDR e}{SHFT *}{CMDR e}{SHFT *}{CMDR e}{S
       HFT *}{CMDR e}{SHFT *}{CMDR x}"
68     1460 return
253    1470 :
156    1480 rem mouse/joy input
10     1490 :
64     1500 if((peek(56321)and16)=0)or((peek(56
       320)and16)=0)then1520
41     1510 goto1500
236    1520 xl=peek(sx):y=peek(sy):xh=peek(sz)a
       nd1
210    1530 x=xl+xh*256
45     1540 x=x-24:y=y-50
```

```
√Σ    IQ.BAS (cont.)

108   155Ø ifx<Øorx>319ory<Øory>199then15ØØ
248   156Ø c=1+int(x/8):r=1+int(y/8)
182   157Ø return
96    158Ø :
163   159Ø rem input decode
124   16ØØ :
129   161Ø gosub15ØØ
1Ø6   162Ø ifc<16orc>24orr<7orr>15then15ØØ
76    163Ø c=c-15:r=r-6:mv=mv(r,c)
6     164Ø gosub187Ø:return
17Ø   165Ø :
79    166Ø rem sound init
198   167Ø :
93    168Ø sc=54272:fori=Øto24:pokesc+i,Ø:next
43    169Ø pokesc+1,3:pokesc+4,38:pokesc+6,248
4     17ØØ pokesc+11,38:pokesc+13,248
53    171Ø pokesc+24,15
65    172Ø return
25Ø   173Ø :
152   174Ø rem good sound
23    175Ø :
37    176Ø x=128:pokesc+8,x:pokesc+4,38:pokesc
      +11,39
129   177Ø gosub183Ø:forlp=1to3
142   178Ø x=x+(x/2):pokesc+8,x:gosub183Ø
15Ø   179Ø x=x/3:pokesc+8,x:gosub183Ø
141   18ØØ next
239   181Ø pokesc+4,38:pokesc+11,38
173   182Ø return
98    183Ø fori=1to1Ø:next:return
1Ø9   184Ø :
1Ø1   185Ø rem click
121   186Ø :
44    187Ø x=128:pokesc+13,24Ø:pokesc+8,x:poke
      sc+11,39
74    188Ø fori=1to3Ø:next
67    189Ø pokesc+11,38:pokesc+13,248
253   19ØØ return
183   191Ø :
245   192Ø rem bad choice
195   193Ø :
1Ø6   194Ø pokesc+4,39
192   195Ø fori=1to2ØØ:next
15Ø   196Ø pokesc+4,38
72    197Ø return
241   198Ø :
112   199Ø rem movement data
2     2ØØØ :
77    2Ø1Ø data Ø,4,6,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø
37    2Ø2Ø data Ø,Ø,Ø,7,9,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø
152   2Ø3Ø data Ø,Ø,Ø,Ø,8,1Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø
91    2Ø4Ø data Ø,1,Ø,Ø,6,Ø,11,13,Ø,Ø,Ø,Ø,Ø,Ø,
      Ø
94    2Ø5Ø data Ø,Ø,Ø,Ø,Ø,Ø,Ø,12,14,Ø,Ø,Ø,Ø,Ø,
      Ø
199   2Ø6Ø data Ø,Ø,1,Ø,4,Ø,Ø,Ø,13,15,Ø,Ø,Ø,Ø,
      Ø
229   2Ø7Ø data Ø,Ø,Ø,2,Ø,Ø,Ø,9,Ø,Ø,Ø,Ø,Ø,Ø,Ø
1Ø5   2Ø8Ø data Ø,Ø,Ø,Ø,3,Ø,Ø,Ø,1Ø,Ø,Ø,Ø,Ø,Ø,Ø
237   2Ø9Ø data Ø,Ø,Ø,Ø,2,Ø,Ø,7,Ø,Ø,Ø,Ø,Ø,Ø,Ø
183   21ØØ data Ø,Ø,Ø,Ø,Ø,3,Ø,Ø,8,Ø,Ø,Ø,Ø,Ø,Ø
194   211Ø data Ø,Ø,Ø,Ø,Ø,Ø,4,Ø,Ø,Ø,Ø,13,Ø,Ø,Ø
139   212Ø data Ø,Ø,Ø,Ø,Ø,Ø,Ø,5,Ø,Ø,Ø,Ø,14,Ø,Ø
2Ø4   213Ø data Ø,Ø,Ø,Ø,Ø,Ø,Ø,4,6,Ø,Ø,11,Ø,15,
      Ø
179   214Ø data Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,5,Ø,Ø,Ø,12,Ø,Ø
67    215Ø data Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,6,Ø,Ø,Ø,13,Ø
162   216Ø :
161   217Ø rem mouse/joy array
182   218Ø :
156   219Ø data Ø,Ø,Ø,Ø,1,Ø,Ø,Ø,Ø
```

```
√Σ    IQ.BAS (cont.)

156   22ØØ data Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø
222   221Ø data Ø,Ø,Ø,2,Ø,3,Ø,Ø,Ø
232   222Ø data Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø
95    223Ø data Ø,Ø,4,Ø,5,Ø,6,Ø,Ø
196   224Ø data Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø
125   225Ø data Ø,7,Ø,8,Ø,9,Ø,1Ø,Ø
2Ø8   226Ø data Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø
174   227Ø data 11,Ø,12,Ø,13,Ø,14,Ø,15
43    228Ø :
25Ø   229Ø rem mouse/joy driver
63    23ØØ :
25    231Ø data 76,78,192,76,246,192,76,252
188   232Ø data 192,Ø,Ø,Ø,Ø,Ø,Ø,128
88    233Ø data Ø,Ø,192,Ø,Ø,224,Ø,Ø
222   234Ø data 24Ø,Ø,Ø,248,Ø,Ø,252,Ø
19Ø   235Ø data Ø,254,Ø,Ø,252,Ø,Ø,248
129   236Ø data Ø,Ø,22Ø,Ø,Ø,28,Ø,Ø
5     237Ø data 14,Ø,Ø,14,Ø,Ø,7,Ø
25    238Ø data Ø,6,Ø,Ø,Ø,Ø,Ø,Ø
15    239Ø data Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø
84    24ØØ data Ø,Ø,Ø,Ø,Ø,Ø,173,25
119   241Ø data 212,2Ø1,255,2Ø8,3,76,152,192
186   242Ø data 173,21,3,2Ø1,193,2Ø8,7,173
112   243Ø data 2Ø,3,2Ø1,2,24Ø,31,32,197
19Ø   244Ø data 192,8,12Ø,173,2Ø,3,141,9
7     245Ø data 192,173,21,3,141,1Ø,192,169
11    246Ø data 2,141,2Ø,3,169,193,141,21
43    247Ø data 3,4Ø,76,246,192,32,252,192
72    248Ø data 8,12Ø,173,9,192,141,2Ø,3
57    249Ø data 173,1Ø,192,141,21,3,4Ø,96
1Ø7   25ØØ data 173,21,3,2Ø1,193,2Ø8,7,173
223   251Ø data 2Ø,3,2Ø1,99,24Ø,223,32,197
1     252Ø data 192,8,12Ø,173,2Ø,3,141,9
247   253Ø data 192,173,21,3,141,1Ø,192,169
182   254Ø data 99,141,2Ø,3,169,193,141,21
133   255Ø data 3,4Ø,76,246,192,162,64,189
165   3328 data 14,192,157,63,3,2Ø2,2Ø8,247
79    3341 data 169,1,141,39,2Ø8,169,24,141
154   3348 data Ø,2Ø8,169,5Ø,141,1,2Ø8,169
24    3358 data Ø,141,16,2Ø8,169,13,141,248
193   3368 data 7,173,25,212,141,11,192,173
174   3378 data 26,212,141,12,192,96,169,1
66    3388 data 141,21,2Ø8,96,169,Ø,141,21
1Ø5   3398 data 2Ø8,96,216,173,25,212,172,11
153   34Ø8 data 192,32,57,193,14Ø,11,192,24
2Ø1   3418 data 1Ø9,Ø,2Ø8,141,Ø,2Ø8,138,1Ø5
4Ø    3428 data Ø,41,1,77,16,2Ø8,141,16
48    3438 data 2Ø8,173,26,212,172,12,192,32
179   3448 data 57,193,14Ø,12,192,56,73,255
46    3458 data 1Ø9,1,2Ø8,141,1,2Ø8,1Ø8,9
12    3468 data 192,14Ø,14,192,141,13,192,162
48    3478 data Ø,56,237,14,192,41,127,2Ø1
231   3488 data 64,176,7,74,24Ø,18,172,13
1Ø3   3498 data 192,96,9,192,2Ø1,255,24Ø,8
112   35Ø8 data 56,1Ø6,162,255,172,13,192,96
2Ø7   3518 data 169,Ø,96,216,173,Ø,22Ø,74
56    3528 data 176,5,72,32,136,193,1Ø4,74
15    3538 data 176,5,72,32,143,193,1Ø4,74
22    3548 data 176,5,72,32,15Ø,193,1Ø4,74
91    3558 data 176,3,32,163,193,1Ø8,9,192
17Ø   3568 data 2Ø6,1,2Ø8,2Ø6,1,2Ø8,96,238
17Ø   3578 data 1,2Ø8,238,1,2Ø8,96,2Ø6,Ø
33    282Ø data 2Ø8,2Ø6,Ø,2Ø8,173,Ø,2Ø8,2Ø8
51    283Ø data 21,24Ø,11,238,Ø,2Ø8,238,Ø
249   284Ø data 2Ø8,173,Ø,2Ø8,2Ø8,8,173,16
241   285Ø data 2Ø8,73,1,141,16,2Ø8,96,-1
```

√Σ

The program listing in this article can be verified with our new CheckSum utility. See page 54.

RUN64

by Doug Cotton

Various routines have been used over the years to allow programs to move from 128 mode to 64 mode without user intervention. With the advent of modified Kernal ROMs (JiffyDOS, RAMLink, and others) many of the methods that work on stock machines have either failed to do the job completely, and in some cases fail all together.

RUN64 is the answer to those users looking to worm their way into 64 mode without having to be concerned with the different Kernal ROMs. The program is presented here in two ways: as a BASIC program that will move to 64 mode and load the program you request, and as assembly language source for ML programmers.

BASIC Notes: The BASIC version uses the ML code produced by the assembly language source. This is found in the data statements beginning at line 660. When you run it, the program will ask for the file name, device number, and file load type (BASIC or ML). The first two parameters should be self-explanatory, but the load type may confuse you. If the file you're loading is itself a small loader (1, 2 or 3 blocks) then it will almost always be an ML program. Likewise, if you usually

```
√Σ    RUN64.BAS

102   100 rem run64.bas (c) 1996 creative micr
      o designs, inc.
168   110 :
19    120 print "{CLEAR/HOME}run64"
26    130 print
198   140 :
81    150 rem f$="filename" : rem filename
232   160 rem dv=peek(186)  : rem device numbe
      r (8, 9, 10, etc.)
87    170 rem l$="a"        : rem load type (a
      =basic, b=ml [,1])
171   180 rem goto 310
248   190 :
235   200 input "filename";f$
160   210 input "{2 SPACES}device";dv$ : if dv
      $="" then 230
42    220 poke 186,val(dv$)
134   230 dv = peek(186)
136   240 print
57    250 print "select a or b"
137   260 print "{2 SPACES}a. load";chr$(34);f
      $;chr$(34);",";right$(str$(dv),len(str$(
      dv))-1)
239   270 print "{2 SPACES}b. load";chr$(34);f
      $;chr$(34);",";right$(str$(dv),len(str$(
      dv))-1);",1"
235   280 get l$ : if l$<>"a" and l$<>"b" then
       goto 280
186   290 print
236   300 print l$;" selected"
206   310 print
146   320 print "going to 64 mode!"
133   330 :
87    340 : rem poke in main ml
153   350 :
177   360 i = 6144
131   370 read d
228   380 if d = -1 then 450
71    390 poke i,d
178   400 i = i + 1
237   410 goto 370
223   420 :
245   430 : rem poke in filename
243   440 :
33    450 for i = 0 to len(f$)-1
119   460 : poke 6356+i, asc(mid$(f$,i+1,1))
232   470 next i
```

```
√Σ    RUN64.BAS (cont.)

103   480 poke 6356+i,0
38    490 :
184   500 : rem poke in device number
58    510 :
57    520 if dv$="" then 570
125   530 poke 186,val(dv$)
84    540 :
249   550 : rem check load type
104   560 :
253   570 poke 6324,0
244   580 if l$="b" then poke 6324,1
138   590 :
241   600 : rem sys to ml
158   610 :
216   620 sys6144
178   630 :
16    640 : rem ml data
198   650 :
32    660 data 32,115,239,160,0,185,22,24
159   670 data 153,0,128,200,208,247,165,186
93    680 data 141,157,128,76,77,255,9,128
2     690 data 9,128,195,194,205,56,48,169
101   700 data 0,141,4,128,120,169,0,141
100   710 data 22,208,32,132,255,32,135,255
119   720 data 169,230,133,1,169,43,141,0
132   730 data 160,169,128,141,1,160,76,248
107   740 data 252,169,231,133,1,32,148,128
77    750 data 32,151,128,32,154,128,162,0
70    760 data 189,159,128,240,6,32,210,255
186   770 data 232,208,245,162,0,189,190,128
4     780 data 240,6,32,210,255,232,208,245
125   790 data 162,0,189,180,128,240,6,32
228   800 data 210,255,232,208,245,173,158,128
94    810 data 240,10,169,44,32,210,255,169
116   820 data 49,32,210,255,169,145,32,210
163   830 data 255,32,210,255,173,157,128,133
213   840 data 186,162,0,189,185,128,240,6
0     850 data 157,119,2,232,208,245,173,158
89    860 data 128,208,2,169,4,133,198,76
21    870 data 157,227,108,149,227,108,152,227
232   880 data 108,155,227,0,0,17,17,68
58    890 data 86,61,80,69,69,75,40,49
135   900 data 56,54,41,58,76,79,65,68
163   910 data 34,0,34,44,68,86,0,13
170   920 data 82,213,13,0,70,73,76,69
104   930 data 78,65,77,69,0,-1
```

load the file with a ",8,1" at the end of the load statement, it's ML. If you're loading a larger file, or a file that you normally load with just a ",8", then use the BASIC option.

Also, if you remove the *REM* instructions from lines 150 through 180 the program becomes a dedicated loader. Just specify the file name and other options within those lines.

Assembly Language Notes: The source code is pretty well documented, and ML programmers should have little trouble figuring out what everything does. Take note of the Buddy Assembler *.off* pseudo-op used a few lines below the *code* label. This adjusts all fixed references within the code that follows it to execute properly at $8000.

The code uses some indirect vectors (*ibv, ibr* and *ibm*), and also switches out the BASIC ROM temporarily to make sure that control is passed back after executing some Kernal code.

To use the routine, just set up a file name at *filename*, put a device number in $ba, set the load type in *sa1flag*, then execute the routine.

```
; RUN64.SRC
; Doug Cotton & Mark Fellows
; (c) 1996 Creative Micro Designs, Inc.
;
        .org $1800
        .obj run64.obj

run64   jsr  $ef73       ; go slow
;
        ldy  #0          ; copy cartridge
-       lda  code,y      ; code to $8000
        sta  $8000,y
        iny
        bne  -
;
        lda  $ba         ; get device number
        sta  dvtemp      ; and store it
;
        jmp  $ff4d       ; go 64
;
code    .byt $09,$80     ; cold start
        .byt $09,$80     ; warm start
        .byt $c3,$c2,$cd,$38,$30 ; cbm80
;
        .off $8009       ; offset code
;
        lda  #$00        ; disable
        sta  $8004       ; cartridge code
        sei              ; disable interrupts
;
        lda  #$00        ; zero out
        sta  $d016       ; VIC control Register
;
        jsr  $ff84       ; initialize I/O
        jsr  $ff87       ; initialize RAM
        lda  #$e6        ; switch in RAM
        sta  $01         ; at $A000
        lda  #<reenter   ; set up return vector
        sta  $a000       ; at $A000 to bypass
        lda  #>reenter   ; BASIC statup during
        sta  $a001       ; initialization
;
        jmp  $fcf8       ; let Kernal finish up
;
reenter lda  #$e7        ; back from Kernal, set
        sta  $01         ; $A000 back to ROM
;
        jsr  ibv         ; initialize vectors
        jsr  ibr         ; initialize RAM
        jsr  ibm         ; initialize memory
;
        ldx  #$00        ; output screen text
-       lda  part1,x     ; to form LOAD statement
        beq  +
        jsr  $ffd2
        inx
        bne  -
;
+       ldx  #$00        ; print filename to be
-       lda  filename,x ; loaded at end of
        beq  +           ; LOAD statement
        jsr  $ffd2
        inx
        bne  -
```

```
;
+        ldx  #$00         ; print device
-        lda  part2,x      ; variable at end
         beq  +            ; of LOAD statement
         jsr  $ffd2
         inx
         bne  -
;
+        lda  sa1flag      ; check secondary
         beq  +            ; address flag for load
         lda  #','         ; type, and print a
         jsr  $ffd2        ; comma and a 1 at end
         lda  #'1'         ; of LOAD statement if
         jsr  $ffd2        ; load type is ML
;
+        lda  #$91         ; print two CRSR up
         jsr  $ffd2
         jsr  $ffd2
;
         lda  dvtemp       ; get device number
         sta  $ba          ; and store
;
+        ldx  #$00         ; put [RETURN]rU[RETURN]
-        lda  keydata,x    ; into keyboard buffer
         beq  +
         sta  $0277,x
         inx
         bne  -
;
+        lda  sa1flag      ; get load type and
         bne  +            ; branch if it is ML (1)
         lda  #$04         ; if not ML, change .A
+        sta  $c6          ; store kybd buffer NDX
;
         jmp  $e39d        ; enter BASIC
;
ibv      jmp  ($e395)      ; initialize vectors
ibr      jmp  ($e398)      ; initialize RAM
ibm      jmp  ($e39b)      ; initialize memory
;
dvtemp   .byt $00          ; device number temp
sa1flag  .byt $00          ; load type (1=ML,
                           ; 0=BASIC)
;
part1    .byt $11,$11      ; 2 CRSR up
         .byt 'dv=peek(186):load'
         .byt $22          ; quote
         .byt $00
;
part2    .byt $22          ; quote
         .byt ',dv'
         .byt $00
;
keydata  .byt $0d          ; [RETURN]
         .byt 'rU'         ; shortcut for RUN
         .byt $0d          ; [RETURN]
         .byt $00
;
filename .byt 'filename' ; name of file to load
         .byt $00
;
         .end
```

CMD • HARDWARE • ORDERS: 1-800-638-3263 • SOFTWARE • CMD

HD Series SCSI Hard Disk Drives

HD-40, 42 MB (Limited Supply)	**$249.00**
HD-500, 500+ MB (Special Edition)	**$399.00**
HD-1000, 1 GB (Special Edition)	**$499.00**
HD-2000, 2 GB (Special Edition)	**$899.00**

FD Series 3.5" Floppy Disk Drives

FD-2000 (800K and 1.6 MB)	**$179.95**
FD Real-Time-Clock Option/Kit	$20.00/$29.00
Box of 10, High Density Disks (1.6MB)	$14.95
Box of 10, Enhanced Density Disks (3.2 MB)	$29.00

JiffyDOS

(Specify computer serial number and drive model)

JiffyDOS C64/SX-64 System	$49.95
JiffyDOS C-128/128-D System	$59.95
Additional Drive ROM's	$24.95

Miscellaneous Hardware

80-column Monitors (Refurb)	CALL
Aprotek 3-Way User Port Expander	$30.00
Aprotek User Port Extension Cable	$19.00
C-64/64-C Computers (Refurb, w/JD)	$89.00
C-64, C-64C Power Supply (Repairable)	$39.00
C-128 Power Supply (Repairable)	CALL
Cannon BJC-4100 Color Bubble Jet Printer	$399.00
Commodore 1541 Disk Drives (Refurb, w/JD)	$75.00
Commodore 1541-II Disk Drive (Refurb, w/JD)	$89.00
Commodore 1571 Disk Drive (Refurb, w/ JD)	$99.00
Commodore 1802 40-column Monitor (Refurb)	$149.00
CMD EX2+1 3-Port Cartridge Port Expander	$34.95
CMD EX3 3-Port Cartridge Port Expander	$29.95
CMD Gamepad/Joystick Controller	$24.95
CMD GeoCable II Print Cable 6ft./15ft.	$29.00/$34.00
CMD SmartMouse (1351 Compatible Mouse)	$49.95
CMD SmartTrack (1351 Compatible Trackball)	$69.95
Inkwell Light Pen Model 170C	$75.00
Monitor Cables	CALL
Mouse Pad	$2.95
MW-350 Printer Interface (ØK/8K Buffer)	$49.00/$60.00
Panasonic KXP-1150 9-pin Printer	$169.00
Panasonic KXP-2023 24-pin Printer	$229.00
Panasonic KXP-2135 24-pin Color Printer	$319.00

Utilities

Big Blue Reader V4.10 (SOGWAP)	$39.00
CMD Utilities	$24.95
JiffyMON-64 (ML Monitor)	$19.95
The Compression Kit '94 (Mad Man)	$39.00

GEOS

Collette Utilities (Handy Geos Utilities)	$19.95
Desk Pack Plus	$29.00
Dweezils Greatest Hits (NewTools2,Stamp,Label128)	$30.00
FONTPACK Plus	$25.00
gateWay 64 or 128 (Specify Version)	$29.95
geoBASIC	$20.00
geoCalc 64/128	$40.00/$45.00
geoChart	$29.00
geoFAX	$39.95
geoFile 64/128	$40.00/$45.00
geoMakeBoot (Makes Bootable copies)	$12.95
geoProgrammer	$45.00
geoPublish	$40.00
GEOS 64 v2.0	$44.00
GEOS 128 v2.0	$49.00
geoSHELL V2.2 (CLI for GEOS)	$24.95
International FONTPACK	$25.00
Perfect Print LQ for GEOS (Laser-like output)	$49.95
RUN GEOS Companion	$20.00
RUN GEOS Power Pak I or II (Specify)	$20.00

Books

Anatomy of the 1541	$12.00
C-64 Science & Engineering	$12.00
C128 Computer Aided Design	$12.00
C128 BASIC Training Guide	$12.00
Commodore 64 Tricks and Tips	$12.00
GEOS Programmers Reference Guide	$35.00
Graphics Book for the C-64	$12.00
Hitchhikers Guide to GEOS	$35.00
Ideas for Use on Your C-64	$12.00
Printer Book for the C64	$12.00
Mapping the C64	$12.00
Simple Internet	$16.95
Superbase - The Book	$15.00

Scanning & Video

Handyscanner	$249.00
Pagefox	$139.00
Video Digitizer	**$189.00**
Videofox	**$89.00**

RAMLink Power-Backed RAM Disk

RAMLink Base Model (ØMB, No RAMCard)	$149.00
RAMLink w/1 MB RAMCard (Special!)	**$199.00**
RAMLink w/4 MB RAMCard (Special!)	**$299.00**
RAMCard RTC Option/Kit	$10.00/$19.00
RAMLink Battery Back-up (Optional)	$24.95
Parallel Cable (RAMLink to HD)	$14.95

MIDI & Sound

Digimaster	$34.95
SID Symphony Stereo Cartridge	$44.95
Sonus 64 Sequencer, MIDI Interface, Cables	$99.00
Sonus 128 Sequencer, MIDI Interface, Cables	$99.00

Languages & Compilers

BASIC 64 Compiler (Abacus)	$17.00
BASIC 128 Compiler (Abacus)	$25.00
Blitz! 64 Compiler (Skyles)	$30.00
Blitz! 128 Compiler (Skyles)	$30.00
Buddy 64/128 Assembler	$39.00
Cobol 64 (Abacus)	$17.00
Fortran 64 (Abacus)	$17.00
Pascal 64 (Abacus)	$17.00
Power C 64 (Spinnaker)	$14.00

Productivity

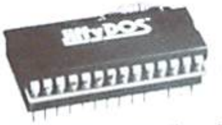
Cadpak 64 (Abacus)	$22.00
Cadpak 128 (Abacus)	$25.00
Chartpak 64 (Abacus)	$17.00
Chartpak 128 (Abacus)	$25.00
Data Manager 64 (Timeworks)	**$16.00**
I Paint v1.5 (128, 80-col., 64K VDC) (Living Proof)	$39.00
I Port v1.54 (128, 80-col., 64K VDC) (Living Proof)	$29.00
Personal Portfolio Manager (Abacus)	$16.00
Pocket Writer 1 (64) Digital Sol.)	$35.00
Pocket Writer 2 (64/128) (Digital Sol.)	$65.00
Pocket Writer 3 (64 or 128) (Digital Sol.)	$70.00
Pocket Planner 2 or Pocket Filer 2 (Digital Sol.)	$35.00
RUN Productivity Pak I, II, or III (Specify)	$15.00
RUN Super Starter Pak 1541 or 1581	$20.00
RUN Works	$20.00
SEC Check Register 128	$29.00
SuperScript 64 (Precision)	$15.00
SuperScript 128 (Precision)	$20.00
Suberbase 64 Version 3.01 (Precision)	$35.00
Superbase 128 Version 3.01 (Precision)	$35.00
SwiftCalc 64 (Timeworks)	$16.00
Perfect Tax 64 (Free '95 Upgrade)	$69.00
Perfect Tax 128 (Free '95 Upgrade)	$79.00
TWS 64 w/Speller (Busy Bee)	$29.00
TWS 128 w/Speller (Busy Bee)	$39.00
TWS Modules (HD/RL/Illustrator)	each $5.00

Telecommunications

Aprotek MiniModem C-24 (C= ready, 2400 baud)	$69.00
Aprotek MiniModem C (C= ready, 1200 baud)	$50.00
Aprotek Modem adapter (C= to Ext. PC Modem)	$20.00
BOCA 2400 Baud Modem	$69.00
BOCA 2400 w/SwiftLink and Cable	$99.00
BOCA 14.4K bps FaxModem	$129.00
BOCA 14.4K w/SwiftLink & Cable	$159.00
BOCA V.34 28.8K bps FaxModem	$199.00
BOCA V.34 w/SwiftLink & Cable	$229.00
Dialogue 128	$29.00
SwiftLink RS-232 Cartridge (Up to 38.4K baud)	$39.95
SwiftLink Modem Cable (DB9-DB25)	$9.95

Games

Atomino	$17.00
Ballistix	$13.00
Blood Money	$15.00
Chomp!	$10.00
Day in the Life of Prehistoric Man	$19.00
Escape Route	$19.00
Heavenbound	$19.95
Island of the Dragon	$19.00
Laser Squad	$13.00
Lazer Duel	**$14.95**
Lions of the Universe	$19.95
Mainframe	$13.00
Menace	$15.00
Navy Seal	$10.00
Rings of Medusa	$16.00
RUN C128 Funpak	$10.00
RUN C64 Gamepak	$10.00
Skate or Die	$13.00
Slaterman	**$14.95**
The Amazing Spider-Man	$15.00
The President Is Missing!	$10.00
The Three Stooges	$10.00
Tie Break Tennis	$16.00
Total Eclipse	$10.00
Ultima V	$17.00
Walkerz	**$19.00**
Wings of Circe	$19.00
Wizardry 5: Heart of the Maelstrom	$24.00



JiffyDOS

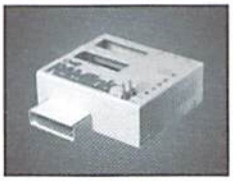
Increase Speed Up to 1500% while retaining 100% compatibility

- Speeds up Loading, Saving, Verifying, Formatting and Reading/Writing of Program, Sequential, User and Relative files unlike cartridges which only speed up Loading and Saving of PRG files
- Built-in DOS Wedge plus 17 additional features including file copier, text dump, printer toggle, and redefinable function keys make using your computer easier and more convenient
- ROM upgrade installs easily into most computers and disk drives. Supports C-64, 64C, SX-64, C-128, 128-D, 1541, 1541C, 1541-II, 1571, 1581 and more.

FD Series

1.6 MB 3.5" Floppy Drives

The FD-2000 disk drives utilize today's latest 3.5 inch disk drive technology. FD-2000's support 800K (1581 style) and 1.6 MB (High Density) formats. Fast and reliable, they support 1541, 1571 and 1581 style partitions, Native Mode partitioning and can actually read and write 1581 disks. FD's feature built-in JiffyDOS, SWAP button and optional Real Time Clock. High capacity, speed and compatibility make the FD right for every application, including GEOS.



RAMLink

Power-Backed Expandable REU

The fastest possible form of storage, RAMLink provides instant access to files and retains data while your computer is turned off. Easy to use and expandable up to 16 MB, RAMLink utilizes the same powerful operating system found in CMD Hard Drives. Unlike Commodore REU's which are compatible with less than 1% of commercial software, RAMLink supports more than 80% of the commercial titles. RAMLink also offers built-in JiffyDOS Kernal, SWAP feature, reset button, enable/disable switch, pass-thru port and RAM port for REU/GEORAM use. RAMLink offers maximum speed, expandability and compatibility with all types of software and hardware including GEOS.

HD Series

HARD DRIVES

SCSI Hard Drive for the C-64/128

HD Series Hard Drives are available in capacities up to 1 GB, are fully partitionable, and can emulate 1541, 1571, & 1581 disks while Native partitions utilize MS-DOS-style subdirectories. HD's connect easily to the serial bus or parallel via RAMLink. Includes built-in JiffyDOS, SWAP feature and RTC. HD's offer superior compatibility with most commercial software including BBS, Productivity and GEOS. And with new pricing, HD Series drives offer the lowest cost per megabyte of any C64/128 storage device.

Shipping and Handling Charges

Use the chart below to match your order subtotal with your shipping zone and method

	Continental United States				
Order Subtotal	UPS Ground	2nd Day	Next Day	AK HI, PR	Canada
$0.01 - $19.99	$3.00	$8.00	$17.00	$12.00	$5.00
$20.00-$29.99	$5.00	$9.00	$18.00	$14.00	$7.00
$30.00-$59.99	$6.00	$10.00	$20.00	$15.00	$9.00
$60.00-$149.99	$8.00	$12.00	$23.00	$19.00	$12.00
$150.00-$299.99	$10.00	$14.00	$27.00	$21.00	$20.00
$300.00-$799.99	$15.00	$20.00	$34.00	$27.00	$25.00
$800.00 +	$20.00	$25.00	$40.00	$32.00	$35.00

UPS C.O.D. add $5.00 (U.S./PR only)

MasterCard VISA **WE ACCEPT ALL MAJOR CREDIT CARDS** DISCOVER NOVUS

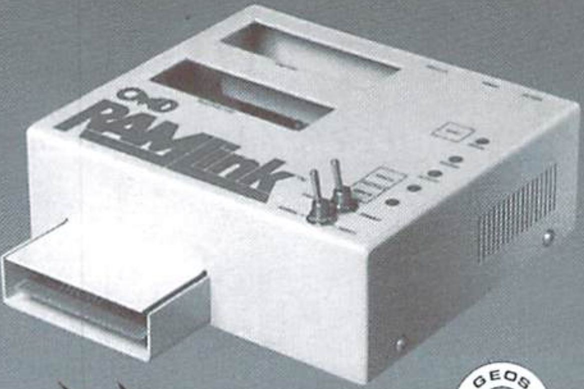
Creative Micro Designs, Inc.
P.O. Box 646
East Longmeadow, MA 01028

Info: (413) 525-0023
Fax: (413) 525-0147

Policies: CMD accepts: Major Credit Cards, Money Orders, COD and Personal Checks. Personal Checks are held 3 weeks. Customer is responsible for shipping charges on refusals. Most items are stock, contact CMD for firm delivery. Returns for merchandise credit only within 30 days with prior authorization. Credits are less shipping, handling, taxes, customs, duties, and a 10% restocking charge. No refunds or credits on opened software. All prices and specifications are subject to change without notice.

Don't Miss the LOWEST Price Ever on CMD's Most Popular Storage Devices

Save up to $60⁰⁰!



Starting at $199⁰⁰

RAMLink w/ 1MB RAMCard	$199⁰⁰
RAMLink w/ 4MB RAMCard	$299⁰⁰
RAMLink w/ 1MB, RTC, Battery	$224⁰⁰
RAMLink w/ 4MB, RTC, Battery	$324⁰⁰

Prices do not include shipping and handling. See our main advertisement on the opposite page for shipping prices and policies.

Add a Battery Back-up plus Real-Time-Clock for just $25.00*

* Only valid at time of RAMLink purchase. May not be combined with any other offers.
This offer valid only in U.S. and Canada through May 31, 1996. CMD reserves the right to adjust prices if necessary due to RAM fluctuations.

To Order Yours, Call 1-800-638-3263

During one of our recent searches for hard disk drives, we were offered a significant number of 40 MB SCSI hard disk drives at a price so low that we just couldn't refuse. Additionally, we were offered some superb pricing on the larger Hard Drives; so low, in fact, that we're offering the HD-500 at a price nearly $1,400.00 lower than the original HD-200 back in 1990. If you've ever considered adding a CMD HD Series hard drive to your system, **now is the time to do it!** These outstanding prices will only last as long as our supply of these specially priced drives. So don't hesitate—add the power and flexibility of a CMD hard drive to your system today!

CMD HD-40 JUST $249⁰⁰

HD-500	540 Megabytes	$399⁰⁰
HD-1000	1 Gigabyte	$499⁰⁰
HD-2000	2 Gigabytes	$899⁰⁰

Prices do not include shipping and handling. See our main advertisement on the opposite page for shipping prices and policies.

CMD CREATIVE MICRO DESIGNS, INC.

Graphic Interpretation

by Steve Vander Ark

GET IT TOGETHER WITH GEOS

I've always been into graphics with GEOS. I enjoy desktop publishing with geoPublish, and get a kick out of customizing my patterns and then messing around with the spray paint tool in geoPaint. Frequently, I'm amazed at the quality of the images which come from geoGIF. Recently, I became the proud owner of a Handyscanner, so my graphic options have really increased dramatically. I've written quite a few articles and columns on the subject of graphics in GEOS, too; from graphics conversion to geoPublish tips. I'd have to say that my favorite GEOS program is geoPublish, followed closely by a number of other graphics-related applications and utilities.

I also really like fonts. Let's face it, fonts are one of the big reasons why GEOS is so cool to use. I have disks full of fonts; some I collected from QLink and GEnie, others that I created myself. I actually enjoy the rather tedious process of clicking on and off hundreds of individual pixels as I try to get a character set to look just the way I imagine it. Along with the graphics applications and utilities, another of my favorite programs is Jim Collette's Font Editor.

There is another variety of GEOS programs which will always keep my interest, sometimes making me waste enormous amounts of time with mouse in hand and eyes straining into the face of my monitor. I'm referring to games, of course. Sure, there are snazzier games for the Commodore outside of the GEOS environment. Just about every GEOS game is a strategy game with nary a role playing game or a shoot 'em up to be found anywhere. But I'm still hooked. Give me geoTiles and Draw Poker and I'm a happy guy.

But when things get gritty—when there's a deadline staring me in the face, like now, for example, those programs get shoved aside in a hurry. When I actually have work to get done, the program I always turn to is geoWrite. Along with the word processor, I always have a few favorite desk accessories and support utilities. See, the kind of writing I do doesn't require any graphics or fancy page layouts. I use whatever font is handy, usually BSW 128, and I put the disks with all the cool games on them as far away from me as possible. It's just me and geoWrite, pounding away to get things done.

That's not entirely true, though; it's not just me and geoWrite. There's also this sort of scruffy-looking legal pad that always sits next to my computer. I use that pad for all sorts of things, from quick ideas for columns to notes on my last conversation with the editor of this magazine. I also keep a scribbled list of things to do and of people to call, as well as details of the project I'm working on at the moment. In other words, that pad of paper is my organizer, my schedule, my address book, and my scratch pad all in one. That's a pretty important tool. A few years ago I learned just how valuable it is to me when I lost it. I was between projects and had just left the pad of paper next to the computer. When my next writing deadline started looming close and I couldn't find the pad, I realized that I was lost. I even had to look in one of CMD's advertisements to get the phone number of *Commodore World* when I needed to call the editor.

It occurred to me, though, that GEOS might just have the tools I needed for all this organizing and the like. Computers do this kind of thing all the time these days. There are several such tools included with the PC version of GEOS, including geoPlanner. Unfortunately, there's no geoPlanner for the Commodore, and as far as I know no one has created anything similar as shareware. But there are GEOS programs that should let me schedule my time, jot notes, track my projects, and much more. The more I thought about this, the more I liked the idea of creating a "project manager" for GEOS by using a suite of existing utilities.

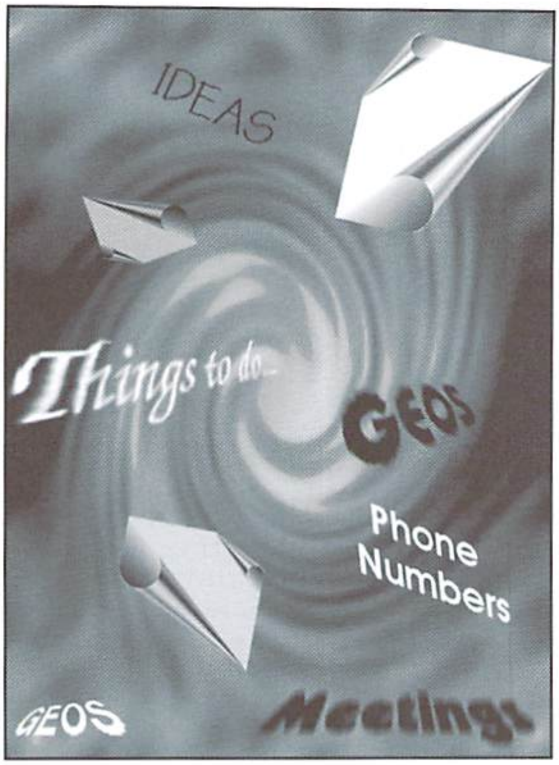
I started by doing a little preparation; by looking at the programs available and then choosing which would work the best. In order to do that, I tried to imagine what I wanted a project manager to do for me. First, it had to be able to keep track of my day, including a to-do list and a place to note things like meetings or calls to be made. I also needed to have a place to jot down an idea or other information that might come up as the day went along. I imagined that I would start the day by creating a nicely-formatted overview of what I needed to do and would then be able to move back to that overview when I needed to.

At first I decided just to fire up the Note Pad desk accessory and leave it at that. I could type pretty much anything on that pad and could then get to it when I needed to by choosing it from the "geos" menu. I gave it a try and was very pleased with the result. I wrote a to-do list there, I added phone numbers of people I needed to call, and I noted any meetings or events on my schedule for that day. Whenever I wanted, I popped up that list and I could see how I was doing. This was certainly a nice solution to my quest for an organizer. As a matter of fact, I almost stopped there. I didn't, though, and the reason was that I wanted to be able to schedule things over several days or even weeks. I wanted to have information for my next deadline stored separately but have it be accessible so I could add things as they occurred to me. Now I know that I could just add pages to my NotePad. But Note Pad pages are empty and small, hardly flexible enough for serious project management. When it came right down to it, I wanted the power of an application.

Let me just pause a second here to clarify what I mean. There is a big difference between a desk accessory and regular application. The difference is mostly philosophical. A desk accessory is an add-on; a supplemental utility which is designed to be called up to do some task that you need done while in the middle of doing something else. A desk accessory is sort of a mini-application, short on power and features, but doing one thing very well. An application is laden with features and capabilities; applications have all the power. Now that doesn't mean that some pretty powerful desk accessories can't be written—Jim Collette's MiniDesk is a good example of a very powerful desk accessory—but otherwise most desk accessories are pretty wimpy. Their big claim to fame, of course, is that they are available from almost anywhere in GEOS without requiring that you close the application you are using at the moment. This makes them extremely useful. But what I was trying to put together had to have more features than a typical desk accessory. What I wanted, I decided, was geoFile.

GeoFile is the "official" GEOS database. Using the mouse to click on a grid and drag to create boxes, I can easily create a screen full of fields (which is what you call the boxes you type information into on a database screen) set up to record all the information about my day. Then I could fill in a new copy of that form for every day. By setting the date as the sort field, I could keep a nice, organized set of daily plans and schedules. I had created enough geoFile screens to know that it would work perfectly for what I had in mind. The only disadvantage would be that it wouldn't have a desk accessory's ease of access, but more on that later.

I hadn't used geoFile for some time, but once I got started, I easily recalled how to set things up on the screen. It's all very intuitive, and with the alignment turned on and by copying fields to create new, identically-shaped fields, I had a workable form in less than a half hour. My first try turned out very well, I thought. I made a series of fields labeled 8:00, 8:30, 9:00, and so on; these were for me to schedule events into. I also included a field for day notes (things like "Issue 13 columns due" get written there) and one for a to-do list. I was at this stage of things when I realized that I had pretty much filled the screen and had not included a call log or a project information section. There was no way that such things were going to fit on one screen, even in 80 columns. A geoFile record is actually larger than the screen itself, so I could have just placed things farther to the right or down a bit. But I wanted to have everything on one screen with no scrolling necessary. I had to make space somehow.



I tossed out all the boxes for different times; I don't ever schedule that much stuff anyway, and just made one box for morning, one for afternoon, and one for evening, each big enough to hold several items. That left me with enough room for everything. It looked great and what's more, it worked. Using the keyboard shortcuts to move around the form made data entry quick and easy. Voilà: I had my organizer.

Rather than try to force geoFile to do things it wasn't designed for, I decided to pull in several other programs which I thought might be useful: geoDex and Calendar, both from Desk Pack Plus. The geoDex program is a desk accessory database which, while not powerful enough to become a full scheduler, can nicely handle storing names and phone numbers and even dial the phone upon request. As for Calendar, it's designed to schedule in whole-day chunks. What makes it handy to have around is the calendar display itself. In fact, if your organizing/scheduling needs are fairly modest, you might be able to keep up using just Calendar. I also kept the Note Pad, since it's quick and easy access made it a great courier between my work in geoWrite and my organizer in geoFile.

All in all, I was pleased with the results. I still wished that I could somehow make geoFile into a desk accessory, though. As it turned out, I could do something very similar to that with a miraculous program called geoWizard. It doesn't actually change any files at all, so geoFile doesn't become a desk accessory. It just starts to act like one. What geoWizard does is provide you with the option of starting a second application while suspending the first, then switching back and forth between the two almost instantaneously. With GeoWizard, I could jump from my organizer in geoFile to my half-finished article in geoWrite at will.

I could still wish for more. It would be nice if I could dial the phone right from my organizer screen, kind of like geoDex does. I do love geoWizard, but having an organizer built as a true desk accessory would be even better. Another handy feature would be an alarm which could be set separately for each item on the schedule. Maybe someone will write the ultimate scheduler/task manager for GEOS one of these days and incorporate some of these ideas. In the meantime, though, this geoFile solution will work just fine. I can keep the odds and ends of my hectic projects on one nice screen, easily accessible, and use that screen to keep myself on track. And the best part is that I don't have to worry about losing that darn legal pad any more!

The SysOp's Corner

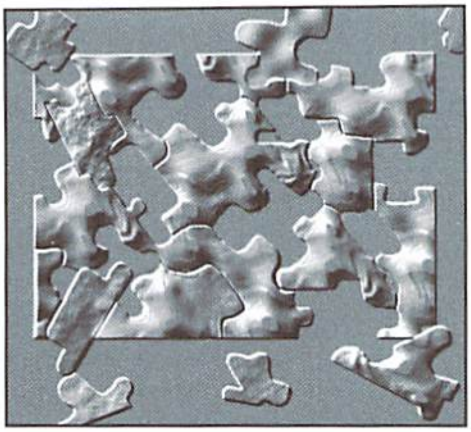
By Max Cottrell

THE MODEM CONUNDRUM

In my first column, I discussed several of the aspects that go into the day to day operation of a BBS. I use my own system as an example for you because if there is a problem to be encountered, I have surely run into it. Here, modems immediately come to mind. Most computer users think of modems as the small box that sits next to the computer with little red lights on it that makes phone calls for you. But what if you want to receive calls? There are a lot of details to consider when using a modem to answer the phone. Modern modems will auto detect what kind of call is coming in. But as we Commodore users know, sometimes there is a little more to what a modem can do TO you than what it can do FOR you. My first modem was an Aprotec Minimodem that would answer the line and take all the abuse that I saw fit to give it. But at 2400 baud, it wasn't exactly a good choice for uploading and downloading the multimedia articles that I write. So, I went out and got myself a state of the art US Robotics. It's the same model that the United States government uses for it's employment BBS.

We've all read what kind of modems we should buy right here in the pages of *Commodore World*. But those are for the purposes of calling out to services, the Internet, and BBSs. The whole world changes when you are at the receiving end. Because you can expect calls from computers that run at 120 MHz (the processor speed of the machine. Stock C-64s run at 1 MHz. A super CPU will run at 20), and send information at a high rate of speed, with a slower system (compared to the high end computers that cost several months worth of wages and a year or two of payments) have to rely on the modem to do the job of receiving and sending the information.

The following are some of the things that you need to know about modems when running a BBS. What happens when modems first link up is called the LAPM (Link Access Procedure for Modems).



This is the noise that the modems make when they connect. A lot of things happen in those few seconds. You'll notice that the higher the modem speed, the longer the modems take to 'talk' to each other. This is the LAPM process in action. Here are some of the things that the modems are talking about. Better put on the 'ol thinking cap...

Modem Speed

Modems vary in price, but as a general rule, the faster the modem, the more you pay. Your choice of modems depends on what you are going to be doing with your BBS. If you are just sending text, then use a 2400 baud modem. But when you get ready to add graphics and downloadable files, then you need a faster modem. It will save transmission time, which will in turn get the user off the phone line faster. Remember: Commodores require a single line BBS. A transmission over a 2400 baud modem that would take 6 minutes will take less than one minute at 14,400 bps. On the information superhighway, speed really is your best friend.

CPS (characters per second)

This is the rate at which a modem will send information over the phone lines. While this relates to the baud rate, it is not the same thing. The figure relates to the phone line condition, the baud rate, and the compression that the modems are using. The rate on a Commodore based BBS can be as fast as on a IBM or a SUN based system. For instance, my Does The Job BBS on my Commodore 64 will have a bps of 210 at 2400 baud, which is pretty good. Line conditions and the modem type will have a great effect on this. A good modem will autotrain (change the rates to match the line condition) as it goes. I have seen the CPS go from 210 to 100 bps over a two minute period while online on my BBS. File type also relates to this. At 14,400 baud, the CPS rates are as follows (assuming a V42 BIS data compression, which is sometimes not possible with a Commodore BBS). Remember that this is under ideal line conditions:

File Type	CPS
Compiled program	3840
Text File	3400
Binary file (program)	2030
ZIP or ARC file	1700
GIF, JPEG	1700

Error Correction

This is one of my problem areas. As baud rates and CPS rates rise, so does the need for error correction. This is one of the areas where my system has problems. Some of the different types of error correction that modems use are highly advanced and can be troublesome to the Commodore owner. Why? Well, there are a lot of reasons, but the buffer within the modems themselves are the biggest factor that I have found. When discussing error control, you need to know that the block of data that is sent/received is called a frame. Error control is done in the following order:

1. Establishment of compatibility
2. Data frame formatting
3. Error correction through Cyclic Redundancy Checking (CRC)
4. Retransmission of bad frames

If a bad frame is encountered, then it needs to be reset. But if your modem is emptying it's buffer at the time, then the file will be bad. Buffer size control is VERY important. As always, when you are the controlling unit in the system, there's a need for doing this exactly right. As an experiment once, I set my error correction commands wrong. Error control relates directly to throughput. The most popular error correction types are:

Type	Maximum Speed (bps)
NONE	300 to 1200
V.22bis	2400
V.32	speeds up to 9600 (will drop down to 4800 on a bad line)
V.32bis	for 4800 to 14,400. Will fall back to lower rates when errors abound
V.42	this is the most popular error correction today. Also supports MNP2 through MNP5
V.42bis	an extension of V.42 that defines the data compression better than V.42 alone
V.FC	newest standard, for speeds faster than 14,400
MNP2 through MNP5	Microcom Networking Protocol (don't ask me, I just write about it.)

As you can see, error correction is a bugger to learn to use correctly on a BBS. I have had the best results with V.42BIS, which seems to be the standard that Commodores like to use.

Data Compression

If the modems establish a V.42 connection, then they try for a V.42 BIS connection. If that's successful, try for a MNP connection. If that works, then they try for MNP5 compression. This is the most important thing in this article: DO NOT allow them to establish a compression protocol higher than V.42! Why? The dictionary is the size of the buffer within the modem that stores the information that is traveling back and forth on the phone lines. If the dictionary is too big, then the data is lost when it is sent to your computer. I've proven this on all four of my high speed modems, using four different brands. V.42bis is more efficient for transferring files from one machine to another.

V.42bis is more efficient than MNP2 through MNP5 compression in part because it dynamically deletes entries that are no longer used. In addition, it works better with files that are already compressed, like ZIP files and 8 bit binary files (program and graphic files), which seem to be compressed already to the modem.

Flow Control

Flow control is required when the modem speed is higher than 1200 baud. Setting the flow control rate higher than the modem connection rate allows for a higher throughput rate. Here are some of the flow control types:

XON/XOFF software requires slow speeds, 2400 baud and lower.

CTS (Clear To Send)requires that your software supports CTS.

RTS (Request to Send)the sister control of above. Both have to be there to be used. Usually called RTS/CTS.

ARQ(automatic Repeat Request) This is a general term for hardware error correction, used by V.42 and V.42bis.

ASYNC (Asynchronous) length between bits being transmitted is varied, and must be signaled for by both modems.

The flow controls are both simple and difficult to understand. As an example, my system will use ARQ when a user wants to download, but won't take a upload using it. I have to switch to ASYNC for a upload, which will work up to 4800 baud. This is related to the buffer sizes within the modems. Confused yet? If not, then read on.

Clock Settings

There are three possibilities here. Always use option one if it is available to you. If you can't set it, then don't worry about it.

1. Your modem sets the clock speed.
2. DTE (data terminal equipment)The computer (your C-64).
3. A loop from one modem to the other is used to time the connection.

Compatibility

There are two main kinds of compatibility: ITU-T (International Telecommunication Union) and CCITT (International Telegraph and Telephone Communications Committee). You will likely be using the ITU-T standards. The other standard is BELL. You will have to experiment with this. As you can see from all of this, the modem it the most important thing in the system. There are, however some things that are general enough in nature that are universal.

Buffer Size

Don't forget this! Your Commodore has a very defined need for a small receiver buffer. Not all of my modems have setable buffers, which limits my maximum upload speed. Downloads are another matter. The modem on the callers end handles that, making it a moot point for you.

Speed

Simply put: faster is better, so spend the money now. In a month you'll be happy that you did .

Navram

I had to bring up something like that, didn't I? NAVRAM is the memory that is user accessible in the modem. This allows you to set things like error correction, buffer sizes, flow control, etc. The AT commands that you send modems access the NAVRAM. Most modems will have at least two memories that you can access for this. A simple command will allow you to view the commands and settings that are stored in it. This also varies with each modem brand. Many come with several templates stored in the memory that you can modify as you see fit. Some modems have a switch that will allow you to have it use the NAVRAM or the factory settings when you turn on the modem.

Verbal vs. Numeric Responses

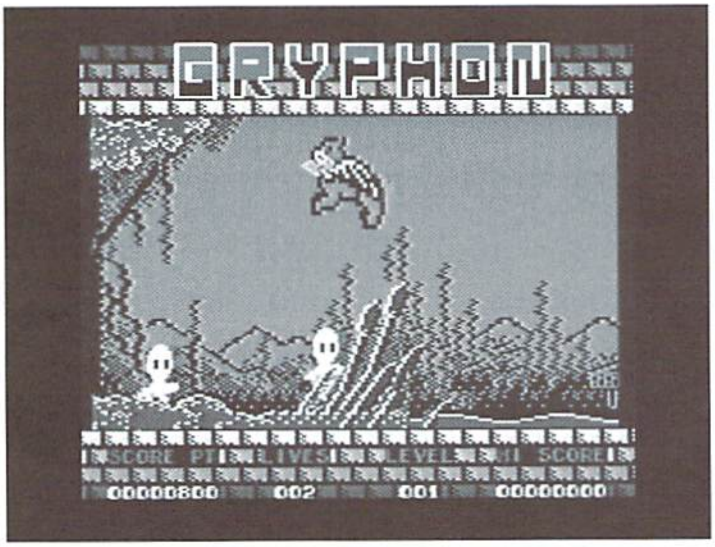
This is the difference in what the modem will tell your computer. When you type in ATZ, your modem will respond with either a 'OK' or a 0 A lot of BBS programs like to use the numeric responses. Does the Job is one of those.

That should be enough for you to think about until next issue, where I'll continue with the saga of finding the right modem. In the meantime, you can contact me at any one of theses addresses:

mcphoto@genie.com
mcphoto@izzy.net
mcphoto@hvcn.org
mcphoto@sojourn.com

GAMES

Test your game skills with Gryphon, The Honeymooners, Apache Strike and Gauntlet II



Gryphon

The Avalon Hill Game Company
Available from SSI

For those of you who may not be familiar with the word "Gryphon", it's a mythical animal typically having the head, forepart, and wings of an eagle and the body, hind legs, and tail of a lion. The Avalon Hill Game Company brings you Gryphon, a Commodore 64 game of "fantasy, arcade and adventure." As you might imagine, this game promises to transport you into a whole new world full of magic, mystery, and danger. The only question is: do you have what it takes to keep your Gryphon alive?

On the left side of your game screen, you'll find gold bars that your gryphon must use as stepping stones in order to cross the poisonous pond. Sounds easy, you say? Well, you're wrong—it's not! I admit, I may not be one of the world's best game players, but one thing I do have is patience. I played this game for quite a long period of time and still found it difficult doing what the book makes sound so easy. First of all, in many games the first level is a sort of training level, giving you plenty of time to master the skills required to get through the rest of the game. In Gryphon, however, they don't give you any warm-up's—the monsters and challenges come at you from your very first step. So be careful!

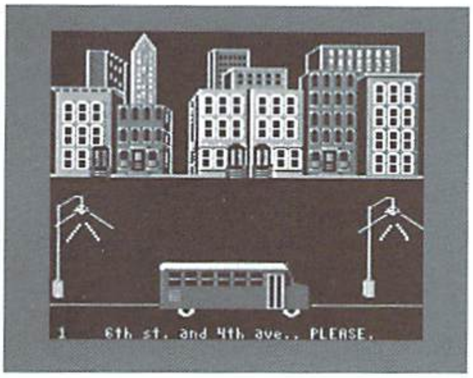
One of the thrilling (yet sometimes frustrating) challenges is that there are monsters to be found everywhere in this game, including spirits that rise straight up from the ground itself! I easily got past the beginning monsters only to encounter the poisonous pond. To cross it, you'll need to pick up a gold bar and drop it into the pond, go back for another gold bar, and so forth, until you've successfully made yourself a little bridge. Picking up and dropping the gold bars may not sound difficult, but it does take a while to master. Once you've made it across the pond, you'll enter the Mystical Woods where more challenges await you in the Surreal Cities, followed by the deadly Darklands. I hate to admit this, but I never even made it to the Mystical Woods. And believe me, I tried!

Unfortunately, as interesting and challenging as the play is, I was disappointed in the graphics in this game. They leave a lot to be desired and detract from the overall experience of such an imaginative story-line. And when you combine below average graphics with almost insurmountable challenges, the result is a game that is not only frustrating, but doesn't hold your attention for long, either.

However, on the more positive side, if you are the type of person who has great perseverance and thinks there is no challenge big enough, and you don't mind the graphics, then you'll probably want to give Gryphon a shot. If you enjoy testing your gaming skills, then you will more than likely find it entertaining. I invite you challenge seekers out there to prove me wrong and make it clear through to the end without enlisting the aid of a gaming cartridge or secret pokes!

-Sherry Freedline

GAMES · GAMES · GAMES · GAMES · GAMES · GAMES
· GAMES · GAMES · GAMES · GAMES
GAMES · GAMES · GAMES · GAMES · GAMES · GAMES
· GAMES · GAMES · GAMES · GAMES
GAMES · GAMES · GAMES · GAMES · GAMES · GAMES



1   6th st. and 4th ave.. PLEASE.

The Honeymooners'
First Row Publishing Company, Inc.
Available from SSI

The Honeymooners', which originated in 1950, introduced the world to Ralph Kramden (Jackie Gleason) and his wife, Alice (Pert Kelton) with a two minute skit performed on Jackie Gleason's "Cavalcade of Stars". Two years later, Audrey Meadows assumed the role of Alice and Art Carney (Ed Norton) and Joyce Randolph (Trixie) were added to the Honeymooners' cast, which was still a skit on the "Cavalcade of Stars". In 1955, The Honeymooners' sketch was turned into its own series, was picked up by CBS, and has since became a legend. Audiences were treated to 39 comedy-filled episodes before the show was cancelled.

The world will never forget Jackie Gleason's infamous expressions such as "One of these days Alice... pow! Right to the Moon!" And now, over forty-six years later, their characters can be enjoyed once again in The Honeymooners' Game by First Row Publishing Company, Inc. The comedy of Ralph, Alice, Ed, and Trixie can be experienced by one to four players on a Commodore 64 or Commodore 128 in 64 mode. The mission of the game is to earn enough train fare ($223) within one weeks time in order to travel from New York City to Miami for the Raccoon Convention.

The game provides four different money-making opportunities; "No Ifs, Ands, or Bus"," Sewer I Will!", "Alice and Trixie Go To Pieces", and "The $99,000 Answer". In "No Ifs, Ands, or Bus", Ralph takes on a few additional bus routes. Assuming the role of Ralph, you must pick up waiting passengers at their bus stops and drop them off at their requested stops. For each discharged passenger, you'll earn between $3 and $12. But, of course, there's a hitch. Each round is assigned a time limit. Each time you play this particular portion of the game, the time decreases. In order to claim your hard-earned bus fare, you must drive your bus back to the bus depot before the time expires. Occasionally, your neighbor Ed Norton boards the bus, adding to your already hectic route.

Next, it's Ed's turn to earn some money for the trip in "Sewer I Will!" At the beginning of this challenge, a map of the sewer system will appear. As Ed, you'll only have a few brief seconds to memorize the map before the gusty New York City breeze takes it from you. Ed must descend into a manhole to repair the many leaks along the sewer lines of the Big Apple. For each repaired leak, Ed earns between $1 and $5. In order to claim his pay, Ed must find his way back to the ladder before the time expires.

"Alice & Trixie Go To Pieces" is my favorite part of The Honeymooners Game. To help earn money for the cause, Alice and Trixie take on jobs building puzzles for display in Morgan's Department Store. Again, a time limit is imposed within which you must properly position the puzzle pieces to earn money. For each properly placed piece, Alice and Trixie earn $1. If they manage to complete the puzzle before time expires, they will earn $25. The puzzles contain pictures of the cast of The Honeymooners'. I have always considered myself to be pretty good at building puzzles, but this game requires quick reflexes and thinking to complete the puzzles within the allotted time. I admit that I found it to be quite a challenge.

If you're successful at earning money in any of the three challenges, you'll be asked if you'd like to participate in Ralph's favorite game show: "The $99,000 Answer". Here you'll have a chance to double the money you worked so hard to earn in the previously completed challenge, by answering a Honeymooners' trivia question. The trivia question gives you a choice of four possible answers. Pick the correct answer and you'll double your money; answer incorrectly and all your hard earned money from the last job goes down the tubes.

All in all, I enjoyed the game. The graphics are cute, but I was disappointed in the lack of sound effects. At the beginning of the game you are treated to the theme song from The Honeymooners', but aside from a few sirens during "No Ifs, Ands, or Bus", that's about it for the sound effects, which in my opinion, detracted a bit from the game. Just a little background music would helped a lot.

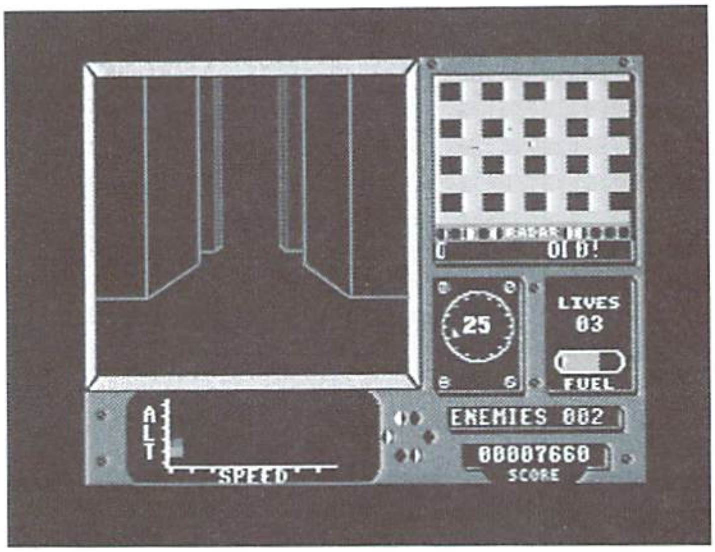
I really enjoyed reading the 12-page manual provided with the game. The appendix includes a fascinating history of The Honeymooners', including how the show inspired other sitcom couples like The Flintstones' and The Rubbles'. The manual also includes pictures from episodes of the classic show. If you are a fan of The Honeymooners', you'll want this game to add to your collection.

- Sherry Freedline

GAMES · GAMES · GAMES · GAMES · GAMES · GAMES
· GAMES · GAMES · GAMES · GAMES
GAMES · GAMES · GAMES · GAMES · GAMES · GAMES
· GAMES · GAMES · GAMES · GAMES
GAMES · GAMES · GAMES · GAMES · GAMES · GAMES

Apache Strike

Activision
Available from SSI



Wanted: A brave Commodore 64/128 owner possessing great amounts of skill, endurance, and courage with an insatiable appetite for danger. Why? To search three cities for their Strategic Defense Computers (SDCs) controlling nuclear arsenals. The recipient of this quest must also possess the know-how to fly the all powerful AH-64 Apache attack helicopter, sporting a sophisticated radar tracking device and ferocious firepower. Only the most advanced high-tech equipment can assist this daring soul in flying at unspeakable speeds through the paths of skyscrapers while eluding the oncoming fire of hostile tanks and choppers. Due to recent threats of total nuclear destruction, the future safety of the world is at risk. Thus, the challenge of Activision's Apache Strike is posed.

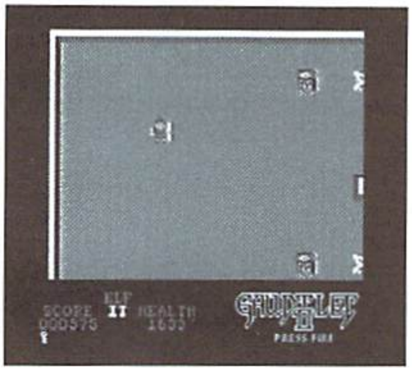
Apache Strike is a one-player game for the Commodore 64/128. A joystick in port 2 is also required. The game begins with the title/selection screen. This screen provides the options of changing the game's difficulty level (a choice of four), viewing the credits, and viewing the high score screens. Once the game's configurations are set, it's time to take to the high skies!

The next screen places you inside the cockpit of the awesome AH-64 Apache Helicopter. At your fingertips is an instrument panel featuring nine different indicators which continually feed you up-to-date information. These indicators include a sophisticated radar tracking device to help you locate the enemies and the SDC's. On the radar screen you appear as an unblinking dot, while the SDC appears as a large blinking dot and any enemies are represented as small blinking dots. As you can see, the radar system keeps you safe from surprise ambushes. The Distance and Bearing Indicators keep you informed of your distance from the SDC. The smaller the number on the Distance Indicator, the closer your are to the SDC. Clock-like in appearance, the Bearing indicator lets you know if you are traveling in the proper direction towards the SDC. When the Bearing Indicator's line is at 12:00, you're heading straight for the SDC; conversely, the line will be straight down from 6:00 if you are traveling in the opposite direction of the SDC. The other instruments keep you informed of the number of remaining helicopters (you get three), remaining enemies (each level has a set number), fuel supply, score, speed, and altitude.

The joystick is used to ascend, descend, move left and right, and to fire bullets. The AH-64 is also equipped with missiles activated by pressing the spacebar. However, keep in mind that each time a missile is fired, five points are deducted from your score. Although it takes three missiles to destroy an SDC, the reward of 2500 points and a successful mission makes it worthwhile. The speed of your aircraft is controlled through the function keys. The speed will need to be adjusted each time you dare to make a turn in the narrow passageways. A help screen is also available by pressing the Inst./Del key to remind you of the various functions of the keys.

From all you have just read, it sounds like a pretty interesting game, right? Well, my expectations were pretty high after viewing Apache Strike's packaging and reading the manual. Unfortunately, this game has one major drawback; the fact that the graphics throughout the levels don't change. The viewing window provides an ongoing display of skyscraper outlines with random appearances of enemy tanks and helicopters. Add in a continual hum from the helicopter and I found that the game lacked the ability to hold my attention or get me into the exciting mood the game promised. However, the sheer challenge of successfully maneuvering my AH-64 through the narrow, maze-like passageways of skyscrapers, buildings, and overpasses was enough to persuade me to withstand the repetitious display. Unfortunately, when left to my devices, the world did not survive. Perhaps, you possess the skills I lack to save the world from total nuclear destruction. Let's hope someone does; I'd like to reach retirement!

- Sherry Freedline



Gauntlet II
Mindscape, Inc.
Available from SSI

Mindscape, Inc.'s Gauntlet II revives your favorite characters from the original game for a sequel featuring an all new assortment of mazes daring to be conquered. All the menacing villains have returned along with a few new ones which are meaner, nastier, and more challenging than ever! Gauntlet II is played on a Commodore 64 (or 128 in 64 mode) by one or two players.

The game begins with the appearance of the character selection screen. The selection of your character is vital to your success in conquering the mazes. You may elect to be Thor, Thyra, Questor, or Merlin. Each character differs in their strengths and weaknesses; Thor (the warrior) has the best shot while Merlin (the wizard) excels in magic. You may need to play the game a few times before you can decide which character is best for you. After you've selected your character, you'll need to pick a color. This option is provided to allow both players to be the same character and easily differentiated from the other. In two player games, both players work together battling the villains. So, of course, you'll get further in the game by enlisting the assistance of a friend.

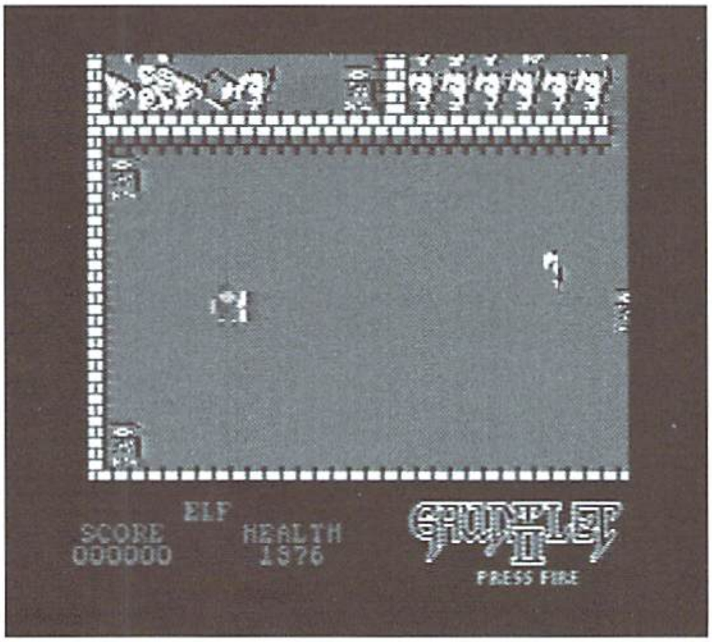
The fun begins as you find yourself in a maze filled with a variety of evil looking characters. The goal is to find your way safely through the maze to the exit door. Each player begins the game with 2,000 Health Points; the game ends when you run out of them. Along your travels, keep an eye out for food, cider, potions, amulets, treasures, and keys. Finding food and cider results in a gain of 100 Health Points, while potions and amulets provide tools to aid in fighting off the enemies such as extra armor, magic power, invisibility, super shots and more. Treasures may be good or bad with rewards of bonus points, potions, or a surprise of death, and finally the keys enable you to unlock locked treasure chests, doors and walls.

The villains include ghosts, grunts, demons, lobbers, sorcerers, super sorcerers, death, acidic puddles, and even "It" and "That". This team of ghouls will hit you, throw fireballs and rocks, disappear, and drain your health. The "It" and "That" characters are interesting; when "It" hits you, all the monsters become attracted to you until you either leave the maze or touch the other player. "That" will also you chase you around the maze. If "That" succeeds in touching you, he'll take away a special power, potion, or 100 health points. You'll need all the strength you can muster, plus any magic you can find, to survive your bouts with these fiendish fellows.

Gauntlet II is a fun and addictive game. The graphics and sound are just what you would expect from a commercial software publisher. The accompanying manual quickly and accurately prepares you for the Gauntlet experience. However, the manual does fail to tell you how many mazes are provided in the game. I know this isn't a significant fact, but it would be helpful to know how far you are from completing the game. The only other downfall to the game is the inability to save games as the mazes are conquered. Losing your life means you have to start again at the very beginning of the game.

So, what are you waiting for? There are ghosts to scare, lobbers to pierce, and sorcerers to curse. When it comes to games, it doesn't get much better than this!

- Sherry Freedline



HARDWARE IN REVIEW

geoKeys

Using an IBM keyboard on your Commodore is easy with this keyboard interface from Germany.

geoKeys

G. Mekes Datensysteme, Birkenstr. 19, 85604 Zorneding, Germany. Available from Geos Users Club; DM 130,00.

Tired of using your C-64 or C-128 keyboard in GEOS? Do you find it difficult adjusting to one type of keyboard at the office and a different type at home? If you answered yes to either one of these questions, then geoKeys may just be the solution that you've been looking for.

What is geoKeys?

Very simply, geoKeys is an interface that lets you use a standard IBM AT-style keyboard with your Commodore 64 or 128 computer. It also includes a disk that contains a setup and test program, and various other utilities that are required to use or configure the new keyboard and interface. Bear in mind that you must provide your own keyboard for use with the interface, as this is not provided. Not providing the keyboard is a good idea, though, as there is a wide variety of keyboards to choose from, and this lets the buyer make his or her own choice.

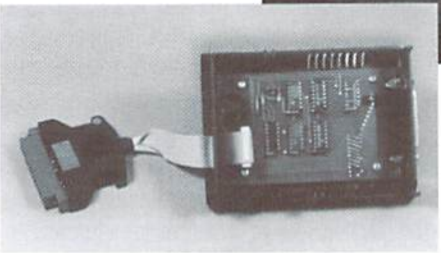
The interface comes housed in a box approximately 5 1/2" x 3 3/4" x 1 7/8". On one end you'll find a standard AT keyboard connector and a ribbon cable that connects to the User Port. On the opposite end the unit has a DB-25 socket that provides for connecting to standard printers (equivalent to a geoCable printer interface).

How Does It Work?

The geoKeys interface hooks into the User Port on any Commodore 64/128 computer. Unfortunately, this doesn't allow the new PC keyboard to be used unless you are running a program that has been specifically written to support the geoKeys interface. In other words, you'll have to use your normal keyboard for everything except GEOS (which is the only program the author provides patches for). In addition, you'll have to load GEOS using your stock keyboard, since the geoKeys patches cannot begin working until after GEOS starts running.

The first time you use geoKeys, you'll need to run though a brief installation process. This is really quite a simple operation, and just involves copying files from the supplied utility disk to your GEOS boot disk; afterwards, you can run a test program to make sure the keyboard and interface are working. Once the installation is complete, the PC keyboard will be available whenever you start GEOS.

What's the Advantage?



As I hinted at earlier, users who spend their days on a PC keyboard and then go home to a Commodore type may find the constant adjustments perplexing. Some users may just prefer a PC keyboard. Others may be attracted to ability to "kick back" with a PC keyboard on their lap. Any of these would be a good reason for getting geoKeys. But there's one more: the EditKeys program supplied with the unit will let you configure function keys to generate keyboard shortcuts—in fact, you can change the definition of almost every key on the keyboard.

In addition, there's a program included called ChangeKeys that lets you quickly switch between keyboard layouts, and a ResetKeys program to switch you back to standard. In addition to all this, the disk also contains an input driver that lets you use the keyboard in place of a mouse.

The Down-Side...

Sadly, the price is the main negative aspect of the product. It sells for around 130,00 DM in Germany, which is just under $100.00 US (at current exchange rates). Luckily, AT-style keyboards are cheap, and you can probably get a good one easily for $25.00 or less. If you consider that the geoKeys interface also doubles as a geoCable interface, it takes a little bit of the sting out of the cost—but it's definitely not something everyone can afford.

The other negative is that there's no English language manual (at least we haven't been made aware of one). Users who have a strong GEOS background should have little trouble, but less experienced users may have trouble getting started.

Overall, though, the product is well designed, and will be a welcome addition for those who want it.

Carrier Detect

By Gaelyne R. Moranec

NOVATERM 9.6

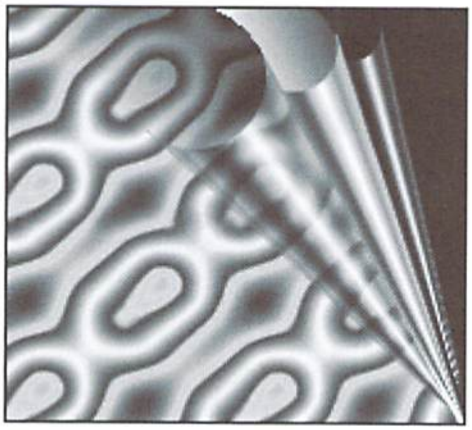
Giving me a new terminal program, especially one which offers something for everyone, is like giving Tim "the Tool-Man" Allen a new power drill or Steve Vander Ark a new GEOS utility to test. As crazy as it may sound, I get excited when it comes to this stuff. A new term program with heaps of new features to play with is the kind of stuff that really makes it fun to write about telecommunications.

Even though I'm a C-128 owner, I've always made it a point to be as non-biased as possible and be familiar with the features that different terminals offer (for either 64 or 128 mode). For my daily use, I tend to go with one of the 128 term programs, but when Novaterm author Nick Rossi offered to send me beta versions of his latest work, I thought I'd give it a look over, but continue to use the other terminals in my arsenal as well. However, this one snuck up behind me and took me completely by surprise. I'm not even sure just when the transition happened, but I soon found myself rearranging the way I perform my daily BBS and Internet access to use Novaterm 9.6. And this was before Rossi made an important discovery and came up with the "pièce de résistance" for 128 users.

Novaterm 9.6, the first ever commercial version of this term program, has something for everyone; from inexperienced users to the modem "jocks" who actually keep track of their CPS rates. And it's not just for Commodore 64 users anymore.

Support Protocols and RAMDisks

Those who have used Novaterm 9.5 are quite aware that this terminal program broke new ground offering Zmodem, which functioned only for downloads, and even then it didn't always work for some users. When I took a look at



Novaterm 9.6, the first thing I noticed was that the file name for the Zmodem protocol no longer ended with "recv". Not only does Novaterm's Zmodem work for both uploads and downloads, it does so reliably. And under proper conditions it will use a streaming mode which continuously sends or receives data until either the transfer is complete, or an error is encountered.

Novaterm 9.6 is the first Commodore terminal program to implement the "-g" (streaming mode) for Ymodem and Xmodem 1k protocols. To use these to full advantage, you'll need a modem capable of hardware flow control and error correction when at speeds higher than 4800 bps. Ymodem-g is the impressive speed demon of the new choices, but it has the distinct disadvantage of aborting if any errors are encountered.

Hand in hand with the new protocols is Novaterm's buffer. The Commodore isn't capable of receiving from the modem and writing to a physical disk at the same time, so it's necessary to use the buffer for file transfers in order to take advantage of the new protocols. Once a barely adequate 9K text-only convention that used what was left of the 64's internal memory, the new buffer is used as a fast RAMDisk, complete with subdirectories (if the user desires). Older versions of Novaterm would install RAMDOS if a RAM Expansion Unit (REU) was detected, and use it as a RAMDisk. This version elegantly combines the function of both buffer and RAMDisk, with the benefit that since it uses custom routines, file transfers are no longer dragged down by RAMDOS. Those without REU's aren't left behind, as Novaterm offers several alternative "memory drivers" to use instead, including the 128's VDC memory, BBGram, GEORAM, and DACC partitions for both the RAMDrive and RAMLink. Any of these devices can be used for the purpose of buffering text, or for file transfers that use the buffer.

The buffer menu has two columns chock full of options, including the ability to view the buffer directory, where files appear with both their disk blocks and bytes listed. Copy options for moving files to and from the buffer give the user a menu list of the disk or buffer files to choose from. This makes it easy to prepare the buffer if you're planning to upload files to a BBS from it, and just as easy to copy downloaded files to a disk drive. Rossi also added many of the file translation methods that are in the Disk Operations menu, like the familiar ASCII-to-PET and PET-to-ASCII translations. Taking it a few steps further, new translations have also been added that will be helpful to Internet users, including a PET-to-UNIX (the ASCII-to-PET translation does UNIX translations at the same time). Also included is the ability to create and decode uuencoded files, from either the disk or buffer menus. Another useful feature is the ability to read text files from the buffer using either the 40 or 80 column fonts.

New Serial Port Support

The first time I checked out Novaterm 9.6, one of the first things that caught my attention was that it offers more serial port drivers than I knew existed! In addition to supporting the standard user port interface and different SwiftLink cartridge configurations (stock, $DF00 and $D700), Novaterm also supports the HART cartridge, which gives the term program the ability to use modem speeds up to 57.6k bps. Additionally, a new cartridge manufactured by Performance Peripherals Inc. called the CommPort is supported. The CommPort is similar to the SwiftLink in that it is a UART serial cartridge capable of 38.4K bps, but it is specially designed to allow BBGram and GEORAM owners to use these devices while modeming. But that's not the last of the new serial port drivers, as Rossi also included support for Daniel Dallmann's UP9600, a modified user port interface that allows modem speeds up to 9600 baud, through the user port (yes, even on the C64). The information for modifying existing interfaces can be found in the comp.sys.cbm FAQ files or on the World Wide Web at *http://rpool1.rus.uni-stuttgart.de/~etk10217.proj.html.*

> "Novaterm 9.6, the first ever commercial version of this terminal program, has something for everyone; from inexperienced users to the modem "jocks"... And it's not just for Commodore 64 users anymore."

New Screen Drivers

Keeping up with incoming text from the modem while not losing any characters in the process is something of a challenge when using the 80 column font in Novaterm. It's been a challenge that Rossi has battled since the advent of 2400 baud modems for the Commodore, and served as an excellent stepping stone to dealing with the faster modems in use today. To keep up with the modem while drawing text on the screen, Rossi has added a "fast-REU" screen driver for use with the 80 column fonts. This driver takes advantage of the REU's direct memory access capabilities, and improves the speed that text can be displayed on the screen without losing characters. For those without an REU, the speed of screen writes has also been increased, using a scroll ahead feature which replaces the former (and sometimes confusing) single, double or triple scroll option.

The "Pièce de Résistance" for 128 Users

If I were to ask Commodore 128 owners to name one reason they would give Novaterm a miss, it would have to be the emulated 80 column fonts. C128 users who have come to appreciate the 80 column mode of the computer have a legitimate complaint, as it's kind of silly to use an emulation of 80 column mode when they have the real thing available. Without this minor "handicap", many 128 users would probably be very happy to take advantage of the newer protocols and speed of file transfers. With the addition of 80 column VDC screen drivers, Novaterm no longer has this "handicap". Yes, if you have a C128 computer, you can have 80 columns and 2MHz when in the terminal mode. The computer must be in 64 mode when Novaterm is used, but 128 users can choose one of two VDC drivers (25 or 28 line) and whenever entering the terminal mode, switch their monitors to 80 column mode. During file transfers or when the user has entered a command that requires them to switch to 40 column mode, the computer switches down to 1 MHz mode and a message is flashed on the screen to remind the user to flip the switch on their monitor.

Novaterm 9.6 has kept its familiar menu interface, but behind the menus there are many new touches that may not be readily noticeable, but make for a very complete program.

In Support of Partitions

Novaterm allows the user to define which devices they want to use for specific tasks, such as having different drives used for uploads or downloads (these can be the same device or two separate ones). This isn't exactly new to the program, but what is new is the ability to define different partitions for storage devices like the RAMDrive or CMD's RAMLink and hard drives. This ability was also added to Novaterm's BBS mode, but set up slightly differently. The BBS mode allows the sysop to pre-define multiple directories or partitions that the caller can access. In this case, though, it requires that the sysop give the directories names that are displayed to the caller.

Other Improvements

Other areas of Novaterm have also been improved. The text editor has block cut, copy and delete functions. The text editor is also used to create and compile script files, which allows more experienced users to automate their online sessions. The instructions for creating script files have been rewritten and are much easier to follow. These take the user through a complete tutorial on creating scripts, and provides detailed descriptions of the many commands it uses.

Keeping it All Tied Together

Something that any veteran Novaterm user will notice immediately is that the terminal emulations, ASCII translations and fonts are all tied together. So, when you change from one emulation to another, all the other associated modules are loaded along with the emulation. This was one of the stumbling blocks users often encountered in older versions because it wasn't always immediately obvious that when you changed emulations that you also had to change the ASCII table or font. The way this is implemented now takes out the guesswork, leaving the user to do more important things.

Making it Easy for New Users

Setting up a terminal program can be a daunting task for a new user, but Novaterm 9.6 makes it easy with a Configuration program which automatically loads the first time the program is used. This takes the user step-by-step through the process of setting up the many options. The Configure program uses both graphics and text to prompt the user about their modem, serial interface, and memory devices. If a user makes hardware choices that don't normally function well together (such as an unmodified RAMLink, SwiftLink and REU), the user is warned and offered the chance to make other choices. For those of us who have made changes to our hardware, there's a kind of "Leave me alone, I know what I'm doing" option. The Configure program auto-detects whether a user is using a C128 or C64, and adds appropriate choices for the memory options as well as asking whether they will be using the computer's 80 column mode. Those who wish to use Novaterm's 80 column emulations are shown the two different font styles the term uses (standard or wide) and asked to pick the one they prefer. Once the font has been selected, the user is given the option to toggle through color choices for both the character and background colors, with the 80 column font shown on the screen. This addition makes it easy to find just the right color combination for reading text online. Once the end of the Configure program is reached, it saves the users answers to disk and loads Novaterm, ready for use.

The manual for Novaterm 9.6 is supplied in printed format and has been completely rewritten. While not condescending in any way, the manual starts off geared towards the new user, taking the reader through using the configuration program, setting up the program and then exploring each menu of the program individually and in depth without being overly technical. Different aspects of modeming, such as uploading and downloading files are covered in detailed, easy to follow steps for each protocol that Novaterm supports. Since many people have more than one computer, the manual also covers how to transfer files using a null modem.

Novaterm: The Cartridge

Novaterm 9.6 doesn't come on a disk, nor does it come as files that need to be downloaded and dissolved; instead, it's a cartridge. Peter Fiset of Performance Peripherals, Inc. designed the hardware and will add Novaterm to the line of Commodore products they carry.

There are advantages to having a terminal program in a cartridge. Long term BBS and Internet users may disagree, but having a cartridge to order and plug in will make it much easier for some people to get started with telecoms. The program is available instantly and any modules needed are loaded quickly, even for those with very slow disk drives. People who have only a 1541 disk drive will have the most to gain, since it will also give them more disk space for other things.

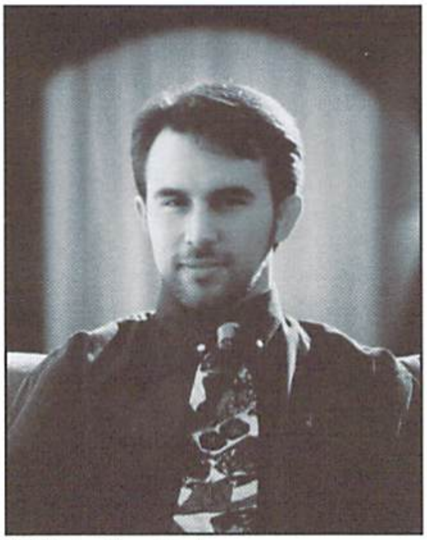
The disadvantage of using a cartridge for the term program is that the cartridge port for many users, is the busiest slot in the computer holding our "extra memory" and our UART interfaces. This adds yet another addition to this port, and in such cases some users will need an expander, which adds to the cost of the product. Another disadvantage is that unless used with an expander that has a switch to make it active (or in active), the cartridge will have to be manually plugged in whenever the user wants to use it, adding wear and tear to the computer. This is my only grumble about Novaterm 9.6, and I don't think I'll be alone in my lack of enthusiastic about the new format.

A cartridge port expander will be offered with Novaterm as part of its "base" package. It will also be offered bundled with other products that PPI carries, such as the CommPort. Those who have registered the shareware version of Novaterm will be offered the product at a reduced price, which wasn't available at press time.

Novaterm 9.6 is the kind of program that makes telecommunications easy and fun. It has something for everyone, aside from the fact it comes in the form of a cartridge. For C64 owners, it gives the opportunity for better and faster communications, and for C128 owners especially, it fills the gaps between other terms that each have great offerings but have been lacking in one feature or another.

Now excuse me, I'm going to go watch another Zmodem upload and marvel at how it wasn't so long ago that this wasn't possible on a Commodore.

Novaterm 9.6 will be available from Performance Peripherals, Inc., 5 Upper Loudon Rd, Loudonville, NY 12211, phone: 1-800-EASY-WEB, or direct from Nick Rossi, 10002 Aurora Ave. N. #3353, Seattle, WA 98133 USA.



Nick Rossi, author of Novaterm

An Interview with Nick Rossi

The Programmer Behind Novaterm

By Gaelyne R. Moranec

Even though Nick Rossi is in Seattle, Washington, and I'm down under in Australia, I was able to interview him by E-Mail and ask him some of the questions I've often wondered when using his Novaterm program. Nick is a 23 year-old programmer for a small start-up company in Seattle, Washington. He received his Bachelors degree in Engineering from Harvey Mudd College, where he also completed several versions of the term program.

Q: *Since Novaterm 9.6 is on cartridge, the users defaults will have to be saved to disk, correct? Are there any other files that will also be on disk?*

A: The configuration defaults, phone books, and function key files all get saved to disk. The cartridge autostarts (it starts running as soon as you turn on the computer) and tries to load the configuration from the first available device number.

Q: *Does it search other devices for the configuration files too?*

A: It will search all the devices until it finds one.

Q: *Have you used the cartridge with expanders and are there any conflicts or problems to be aware of?*

A: I've been using CMD's port expander for a while without any trouble.

Q: *Will previously registered users have to purchase 9.6?*

A: The cartridge will be available to registered users at a reduced price. The exact price hasn't been nailed down yet.

Q: *Can Novaterm 9.6 be purchased without the cartridge?*

A: No. Call it "advanced copy protection".

Q: *With the streaming protocols, how does it "know" if an error has occurred?*

A: When there's a bad CRC, sync error, etc. Any of the typical Xmodem-style errors that would normally cause a block to be re-tried, instead abort the transfer completely. That's the standard with the -g protocols.

Q: *The opening screen for Novaterm has had a different phrase over the years. Some are about Commodore computing, or about computing in general. Do you ever get feedback from users about these? When did you start this?*

A: Every now and then someone mentions something. It's mostly for my own personal amusement. This time, however, I put in a little note about fighting government interference on the Internet, in light of recent attempts at ridiculous, pathetic, First-Amendment-violating regulations.

Q: *What do you see in the distant horizon for Novaterm?*

A: Right now I'm going to wait and see what kind of response the cartridge gets. If there's a good response, and if there are new features it really needs, then I'll consider yet a new version. For now, though, there are other commitments (outside of computers) in my life that I'd like to spend time on. I know there's a lot of activity right now toward developing a SLIP interface for the Commodore, and while I'd love to get involved in that—it would be perfect for Novaterm—I'm going to abstain and watch what others come up with.

Q: *When did you first begin using a Commodore?*

A: 1984 (I was 12). Prior to that, I had a TI99/4A. I never thought the jump from a 16-bit computer to an 8-bit computer could be so incredible.

Q: *When did you start modeming and what was considered "state of the art" at the time?*

A: My first modem was a Commodore 1660 that I got for Christmas 1985. At that time, not even the C128 or Amiga were out yet, so the C64 was IT... at least, I thought so!

Q: *How old is Novaterm? When did you start it?*

A: I started it that same Christmas I got the modem. I remember the modem came with a term program called "Common Sense" which I hated. Xmodem downloads would not work, and it had no Punter protocol. I decided then that I would have to write my own program, as I had done to accomplish countless other little tasks. I was enthralled enough by bulletin boards that I also started writing BBS software at the same time. What can I say? I was very ambitious. The BBS eventually went on-line, from October 1986 until August 1989. I let the software go after that.

Q: *Do you recall what your first goals for the program were? What were its key features then?*

A: I wanted a program environment that was organized sensibly and had a bit of aesthetics to it. None of the term programs available at the time had the features I wanted, and some of them looked downright horrible. I also realized that in order to support different protocols and modems, I'd have to separate them into loadable modules. The other programs tried to cram everything into memory at once, and I knew they'd have nowhere to go once they used up the 64K. Of course, the modules concept turned out to be instrumental in allowing Novaterm to support a wide variety of features.

Q: *Was Novaterm always your own program? Which came first, the Novaterm written for the 128 or yours, and do they have anything in common?*

A: I started writing Novaterm early in 1986. In those early years, I worked on it with a friend named Randy Parker, who provided a lot of ideas. Eventually, his interests took him elsewhere, and I continued with the program. That was before I had access to the national on-line services (like QuantumLink and GEnie), so I had only spread the program around locally. The following year, someone on the east coast wrote a 128 term program and also called it "Novaterm". Because my Novaterm was not on any on-line services at the time, the author of the other one had no way of knowing he had chosen the same name. Not long after he released his program, he gave up on Commodore computers altogether, so there was never much of a problem with the name, other than some occasional confusion by various people. But the two programs have nothing to do with each other.

Q: *What versions of Novaterm were completed while you were in college?*

A: I finished Novaterm 8.0 just before going off to college. During college I released versions 8.2, 9.0, 9.1, 9.2, 9.3, and 9.4, at a rate of about one version per year. I finished version 9.5 after I had been out of school for awhile.

Q: *How difficult was it to study and work on Novaterm at the same time?*

A: Work on the program went slowly because I went to a demanding school and I had a lot of other things going on.

Q: *What is your computer set up today?*

A: I have a JiffyDOSsed C64, two 128's, 2 1541's, a 1571, a 1581, SwiftLink, HART cartridge, REU, BBGRam, RAMDrive, RAMLink, a 2400 baud modem, and a 14.4 modem.

Q: *When did you make the transition from coding on the C64 to using a cross assembler? And why?*

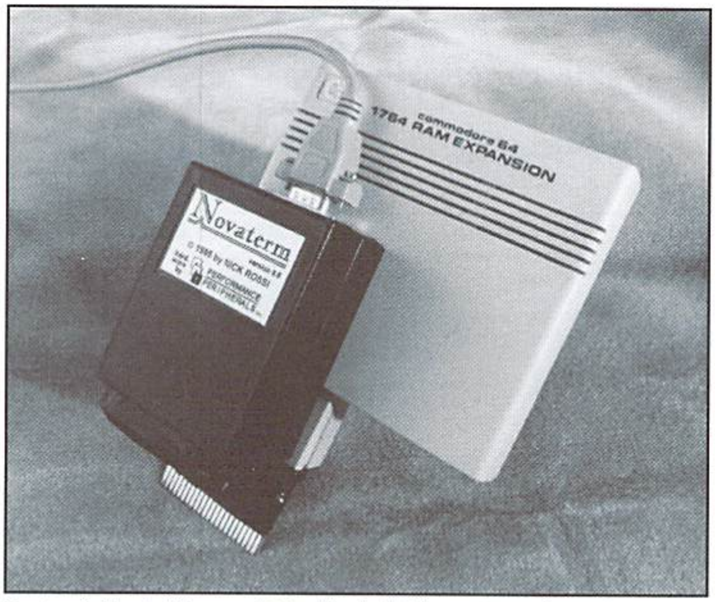
A: I got my first IBM clone just before going to college. It was an old 8 MHz XT (not old by the standards of those days). I picked up a null modem cable, and soon I was editing and assembling source code on the PC and transferring it to the 64. I went that route because it was much faster than using the Commodore. I couldn't afford any real peripherals for the Commodore (being in college) so the PC was the perfect choice. Since then, I've just transferred my stuff along as I upgraded my PC.

Q: *What cross assembler do you use?*

A: I found the C source code to a German-written assembler. I don't recall what it was called. I took the source code, taught myself C, and started changing it to make it easier for me to use. I've added lots of features to it since then, mainly just the things I needed when I needed them. It's been very useful having the source code to the assembler; it makes code-writing that much easier.

Q: *Would it be a good guess that you have a Novaterm script for moving the files across to the Commodore?*

A: Actually, I just use the Zmodem auto-download feature. All I have to do is initiate the transfer from the PC.



Novaterm & Friends: A Novaterm 9.6 Cartridge, SwiftLink, 1764 REU and PPI 3-slot expander.

Q: *How many hours a day do you spend programming?*

A: Anywhere from 6 to 16 hours. I have a real job, which takes priority, and it depends on which days I feel like working on Novaterm. That's the main reason why it has been about one and a half years since the last version.

Q: *What are your reasons for changing Novaterm from shareware to a commercial product? Do you have any comments on the subject of shareware?*

A: Novaterm actually did fairly well with shareware—I haven't heard of another C64 program that has done as well with it. But there's a lot of time invested in the program. I've always been the kind of person who has too many interests and not enough time for them all, so in order to justify spending the time on Novaterm, I needed the prospect of making some money from it. I admit that freely; I'm a dyed-in-the-wool capitalist, and I think that creating a product of value and being rewarded for it is one of the noblest efforts a person can undertake. I know that older versions of Novaterm have been of value to people, and the new one is so much more powerful, it's going to be worth more to them.

Novaterm 9.6 Sources

Performance Peripherals, Inc.
5 Upper Loudon Rd
Loudonville, NY 12211
Phone: 1-800-EASY-WEB.

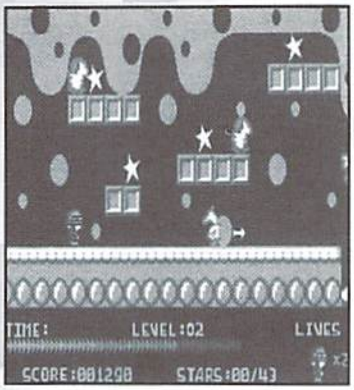
Nick Rossi
10002 Aurora Ave. N. #3353
Seattle, WA 98133 USA.

MORE NEW GAMES

EXCITING NEW PRODUCTS AVAILABLE FROM CMD IN 1996...

THRESHOLD PRODUCTIONS
presents

SLATER MAN



Armed with an unlimited supply of ammo, you set out to retrieve all the stars and terminate or avoid the creatures who try to stop you, while humming the catchy tune featured in this game. SlaterMan features five unique screens for each level, passwords for moving to a given level, plus the ability to enable both music and sound, just music, just effects or silence. SlaterMan is a well designed game promising a great escape from everyday stress; but best of all, you can play it without reading volumes of instructions!

SLATER	SlaterMan	**$14.95**

WALKER'Z

Walker'z are genetically engineered life-forms programmed to reconstruct dying planets, and turn them into class M planets which can support human life. Walker'z do not require oxygen, can go weeks without food and water, are 10 times stronger than a man, and have a life span of more than 200 years.

In the year 2101, Walker'z are sent to the planet Talia, which is about to implode and create a black hole. Only 72 hours remain when the Walker'z land on the planet. However, shortly after they take up position, a freak magnetic storm completely erases their memory, causing them to run amok across the planets surface.

You watch in horror from your sattelite as the Walker'z wander aimlessly; running the risk of falling into chasms, power generators, or off the edge of cliffs! Realizing that you and all of the Walker'z could be destroyed by waiting for technicians from Earth, you try to find a way to save the Walker'z until help arrives. By diverting power from the satellite's shields, you can create a shield on the surface, large enough to save one Walker at a time. With everything on the line, your reflexes and concentration must be keen!

WALKERZ	Walkerz by EBES	**$19.00**

COMING IN WINTER '96
SWORD of HONOR

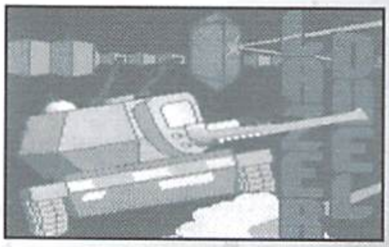
The dynasty sword, known as the Sword of Honor, has been in Shogun Yuichiro's family since the beginning of time. Tradition says that a Samurai's sword is his soul and that no one else can touch it, which is why the Yuichiro sword has been guarded for centuries.

The power-hungry damion, Toronaga, ruler of northern Japan, would do anything to defeat his rival, Yuichiro, ruler of southern Japan. To further his quest to rule Japan, Toronaga has stolen the Yuichiro sword. Finding that his sword has been stolen, Yuichiro calls you, his most dangerous ninja warrior, to reclaim his sword and family's honor! Despite your power, you must beware—rumor has it that the evil damion has mysterious powers!

If the story line didn't get you, the spec's on this phenomenal game will. Try 2.5 megabytes of compressed data - that's right, four disk sides or one 3.5" disk! Outstanding graphics—over 90 colors on screen! Outstanding Sound! It also features: lethal weapons, vicious hand-to-hand combat, and awesome gameplay! And if that's not enough, there's even a special 3.5" version for1581, CMD FD, HD and RAMLink owners! So keep your eyes open for the Winter '96 release of this outstanding action game!

THRESHOLD PRODUCTIONS
presents

Lazer DUEL



Fusion powered laser tanks firing through a field of geometric shapes that deflect or absorb each laser blast make up the field of play in this game of skill. Only those who have mastered the technique of banking shots in the direction of their enemy will survive! Single or Two player, Lazer Duel offers 40 screens and 80 levels—each victory bring's a greater challenge, making it difficult to beat the game!

LAZERDUEL	Lazer Duel	**$14.95**

To Order CALL CMD
(800) 638-3263
See our general ad for shipping details

GEO PROGRAMIST

READBYTE

by Maurice Randall

There may come a time when you are working on an application and find a need to do file accessing for various reasons. Maybe your application will have the capability to display text files to the screen, or maybe you just need a method for reading data from a file. GEOS provides us with many different ways for reading files, so we'll take a look at some of these.

Opening A File

Most of you are familiar with opening and closing files from BASIC. You might also be aware of the problems that can arise if a newly created file is not properly closed.

The 64 and 128 both have kernal routines for opening and closing files, and the machine will keep track of those open files for you. These kernal routines are known as 'OPEN' and 'CLOSE'. These are the routines that BASIC will access when reading or writing a file. You would also access these from machine language. At the same time, the drive will keep track of an open file until it is properly closed. So technically, the computer is opening a file and so is the drive. Also from machine language, you can bypass the OPEN and CLOSE routines and access the drive directly to tell it to open and close the desired files. This way, the kernal routines are not managing the files, only the drive is. Your own software routines would now be in charge of keeping track of which files are open.

In GEOS, we take a completely different approach. A file is never really opened as far as the drive is concerned. The opening and closing of files is completely handled within the computer. Most of the time, it will be your own program that must keep track of the files that are open. This will be easier to understand as this article progresses.

Whenever you read or write to the drive, it is most always done in blocks of 256 bytes. You can't just read one byte from a sector. You must read the whole sector into memory and then read the desired byte from memory. Reading a sector is very simple. Just load r1L with the track, r1H with the sector, and r4 with the address in memory where you wish the sector to be placed. Then call GetBlock to do the work for you. This is how the code would look to read in track 18, sector 1 and place it at diskBlkBuf:

```
LoadB r1L,#18
LoadB r1H,#1
LoadW r4,#diskBlkBuf
jsr GetBlock
```

Now you will have all 256 bytes from that sector in memory and you can manipulate it however you wish. If you want to write it back to disk, a simple call to PutBlock will do this for you. PutBlock expects the same registers to be loaded with the track, sector, and memory address. The values you originally placed in r1L, r1H, and r4 were not altered when the block was read in, so there is no need to load those values again unless you did something in the meantime to change them, or you want to write to a different sector or from a different point in memory.

Reading One Byte At A Time

Whenever I need to read bytes from a file, a favorite GEOS kernal routine of mine is called ReadByte. This allows you to read one byte at a time from a file. Didn't I just say that you can't read one byte at a time? Well, yes and no. When you call ReadByte, it will actually read the whole sector into memory and then place the first data byte (actually the third byte of the sector) from that sector into the accumulator for you. The next call to ReadByte will put the second data byte into the accumulator and so on. Only the first call will actually access the drive, while each additional call will read from memory until you reach the last byte of the block, at which point another read access will occur to fetch the next block in the chain.

ReadByte will do all the hard work for you, so all you have to do is preserve a few pointers in between calls. Those are r1L, r1H, r4, and r5. These pointers must also be properly set up before the first call to ReadByte. So, let's say we want to read the bytes from a file. Find the starting sector from the file's directory entry and place this track and sector into r1L and r1H, respectively. Then load r4 with the 256 byte page of memory that you want ReadByte to work with, which would normally be diskBlkBuf. Then load the two bytes at r5 with zero which will tell ReadByte that the sector is not yet in memory. Let's look at the whole routine and then discuss it some more:

```
ReadExample:
  LoadW r6,#filename
  jsr FindFile
  txa
  bne 90$
  MoveB dirEntryBuf+1,r1L
  MoveB dirEntryBuf+2,r1H
  LoadW r4,#diskBlkBuf
  LoadW r5,#0
 10$
  jsr ReadByte
  cpx #0
  beq 20$
  cpx #BUFFER_OVERFLOW
  bne 90$
```

```
   rts
 20$
    ... ;do whatever you
    ... ;want with the data
    ... ;at this point.

   bra 10$
 90$
   jmp DoError

filename:
   .byte "MyFile",0
```

The first thing you need to do, as the example illustrates, is to find the desired file on the disk. Of course, we need to also know what the filename is that we're looking for. If this is always going to be the same file, our program can be coded with the filename, as in the example. Otherwise, you could prompt the user for a filename or allow the user to select a file from a directory listing or a dialogue box. In any case, the GEOS routine FindFile will locate the file in the directory and place the directory entry into dirEntryBuf for us. Now we know where the starting track and sector is because it is always the 2nd and 3rd bytes of the directory entry. The example then sets the parameters that ReadByte needs and the reading begins.

Always check for disk errors. FindFile will return an error in the .x register if one occurred. As long as .x = 0 then no error occurred and we can safely assume that the file exists. If an error occurred, a branch to 90$ will occur and then a jump to our own error handling routine that will inform the user of the error.

Likewise, ReadByte will also return an error in .x, so as long as .x = 0 there will be a valid byte from the file in the accumulator. The error known as BUFFER_OVERFLOW has a different meaning with ReadByte than it would normally have if generated from some of the other GEOS routines. When this error occurs, it simply means that the end of the file, or the end of the chain of blocks has been reached. At this point, our example ends.

Using ReadByte is nice because it doesn't necessarily read until the last byte of the last sector in the chain. It pays attention to the first two bytes of the last sector. The first byte will always be a zero and the second byte points to the last byte of the file. This is where the reading stops. The next access to ReadByte will generate the BUFFER_OVERFLOW error.

Reading VLIR Files

This sort of file reading works fine with a sequential type GEOS file or a non-GEOS file. But what if the GEOS file is a VLIR file? In this case, the file is stored in a series of one or more records, with the pointers to the starting block of each record stored in a single sector known as the VLIR index. You could load this index into memory and have access to each of these pointers, or you could have GEOS manage the record handling for you. VLIR files are fairly commonly used in GEOS. GeoWrite data files are stored this way. Each page of the file is stored in a separate record. This makes it easy to jump to any page of the file. GeoPaint also stores it's files in the VLIR format. That is how GeoPaint can easily move up and down through the file as you use the scrolling tool. There isn't enough memory to load an entire GeoPaint page into memory, so this method is a good way around the problem. We will study VLIR files in the next issue.

BASIC INSTINCTS

By David Pankhurst

As the newest member of the *Commodore World* writing crew, I'd like to use this article to introduce myself and talk about this column's purpose and focus. My name is David Pankhurst, I live in Winnipeg, Manitoba, Canada, and I've just taken over the column duties from Mr. Barker. This is not my first foray into computer magazines; besides having written a column for the *Compute Magazine* disk until its demise, I've worked on various programming and technical writing projects for other magazines in the Commodore field, as well as my own programming projects.

Like many of the readers of this magazine, I went through a series of computers before settling on the Commodore 64. In the early eighties, there were a lot of choices but little compatibility. This meant quite an investment, not just of money but of time spent learning a new system each time a small company closed. My first real computer was the Timex-Sinclair 1000, with 2,048 bytes of memory for programming. Next came the Radio Shack MC-10, a computer few people may remember (and even fewer bought). Then, my pride and joy, my C64. Although I've worked on other computers, I can honestly say that I've never invested so much time learning, or learned so much from any other system. Happily, I can focus on it again in this column.

From its Humble Beginnings...

But enough about me—now the column. BASIC as a language has a long and varied history. Invented about forty years ago, it stands for Beginner's Allpurpose Symbolic Instruction Code. Although the acronym is somewhat contrived (all programming languages can be considered symbolic instruction codes), the main point is that it was intended for beginners, and that was how it was taught at universities. It became one of the 'classic' languages developed in the fifties that has survived the years well, along with others such as COBOL, FORTRAN, and LISP. For the IBM compatible world, BASIC was there right at the beginning; the first product Bill Gates of Microsoft fame created was a BASIC interpreter. Since then, BASIC has accompanied the release of every major computer on the market. With this popularity came more power and versatility as commands were added and enhanced, until BASIC of today stands as robust and all-purpose as any other language.

What about the C64 and C128 versions of the language? Their BASIC has a very rich command set, allowing you to work with the computer easily and concisely (although a major exception would be the POKEing needed to get the C64 sound and graphics running). The C128 comes with greater functionality, having a BASIC that holds its own even today. And unlike other computer systems, it's built in, ready to run on startup.

That's not to say there aren't disadvantages with the language. For instance, many BASICs are interpreted, rather than compiled. The text that we understand as BASIC has to be converted into commands that the computer understands; this translation can be done once (compiled) or every time the computer program is run (interpreted).

The disadvantage comes with all the checking interpreters have to do, trying to catch code mistakes such as POKEing to a negative location, or writing out of bounds on an array. Conversions add their penalty too, as the computer takes our version of keywords, numbers, symbols, and such, and converts them to computer representations (floating point numbers are notoriously slow examples). This all adds a time penalty that's felt when run.

Additionally, BASIC can't do everything well. For one thing, it's not efficient with memory. A Commodore BASIC program resides in memory in a semi-compacted form, taking up more room than if it was compiled. Working with input and output can be awkward—look at screen input and relative files. Complicated math (such as complex number mathematics) has to be written in, rather than included as part of the language. And the simple commands are fine for simple tasks, but if you wish to vary anything, the task becomes more difficult. One example is printing at the bottom right corner of the screen without forcing a screen scroll; it can be done with POKEs, but not by the PRINT command alone.

Basically, a Great Language

That's the bad news. Now the good. No one computer language can be all things to all people. Where BASIC is weak, other languages excel; but the reverse is also true. For one thing, BASIC is concise. Take a simple example from C programming:

```
#INCLUDE <STDLIB.H>
#INCLUDE <STDIO.H>

INT MAIN() {
{
  PRINTF(“Hello, World!”);
  RETURN (0);
}
```

The same program in BASIC is

```
PRINT"Hello World!"
```

Not only would you get this example running faster than the C version, but you would have less maintenance to do. If fact, you wouldn't even consider saving this example, it's so easy to rewrite. BASIC assumes a lot about the system, allowing you to put together a standard program fast. In C the computer assumes less, so you're forced to tell it where the data is going, how to set it up, and even split the program into sections (MAIN() for instance, isn't a command, but a direction to the compiler that this is the main part of the program). In C this flexibility has its advantages, but BASIC, with lesser flexibility, can produce smaller programs.

Secondly, BASIC adds strong error checking. If you overflow an array, BASIC tells you. Same with dividing by zero, or writing to a device that is not open. With a significant amount of time spent debugging software (in the programming profession, it's been estimated that 50% of the time on a project can be spent debugging software that only took 17% of the time to write), the difference adds up. Error messages are a blessing in programming. Try working with a language that tells you nothing when an error occurs, such as machine language, and you'll be convinced of the time it saves you.

Third, BASIC on the Commodore computers is interpreted. This is a disadvantage in speed of execution, but an advantage in programming. While compiler users are writing a program, running it through a compiler, testing it, and repeating the cycle, an interpreter user can test out several options, running the results immediately. This is especially noticeable when you're tweaking a program. Little changes take as long to compile as major ones, but take no time at all to edit and run with an interpreted language.

BASIC also benefits from its beginner roots. No one has problems learning what PRINT, RUN, or LOAD does. But many languages have problems in that area. Take a look at some cryptic C keywords: "EXTERN", ">>=", and "STRUCT" (I really like C—honestly I do— its just that it is such an opposite to BASIC in so many ways that it makes for great examples). And just type PRUNT or PRIN, and you'll spot the mistake immediately, but not so easily in other languages.

What's in Store

This only touches on a few of the advantages of this language. I'm delighted to have this column to present ways to make BASIC do things you need to do. In line with that, next month we begin by looking into routines to perform common tasks. The focus of this column will be how BASIC programming can solve common problems; however, this focus is malleable. If enough readers (or even one interesting and articulate one) wish to look at a particular BASIC programming task, I'll consider it. If there are things in the Commodore programming arena you've wanted to do, but weren't sure about, or if you just want another viewpoint, send me a line to the address located at the end of this column. Of course, this column hasn't the space to write whole programs, but often only a small chunk of code is needed to solve a specific problem. In addition, I hope to cover some of the thornier programming challenges, such as higher math, sorting and such; things that no one really wants to learn to get a job done, but sometimes needs to. Hope you enjoy it, and I'm looking forward to this opportunity to help us all explore our BASIC Instincts!

David Pankhurst
c/o Commodore World Magazine
P.O.Box 646
East Longmeadow, MA
01028-0646

Peripheral Vision

By Jim Butterfield

THE CASSETTE CONNECTION

The cassette tape peripheral, or Datasette, as Commodore called it, seems to be a useless leftover from the early 8-bit days. Yet many people still use it, especially in Europe, and the connector is still there and useful in many ways. It has two power connections; one was for the Datasette's electronics, a regulated 5 volts supply. The other was switchable power to run the Datasette's motor. For this purpose, it has higher voltage and current rating. There are two input lines. One is connected to the Datasette's PLAY button, and calls for power to be supplied to the motor. The other is the data line itself, feeding a digital signal in from the tape. And of course there is the output line, which writes to tape.

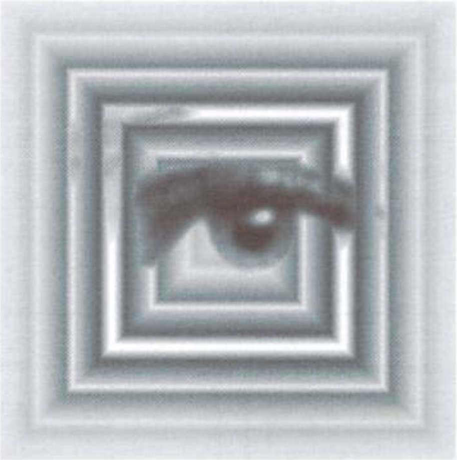
That accounts for six pins—two for power, two for input, one for output, and of course a ground line. The edge connector seems to be a twelve-pin job, but that's an illusion: top and bottom connections are identical. See Figure 1 for the pin configuration.

Levels of Interface

You can interact with the cassette port at several levels. Each level involved a different amount of hardware/software involvement.

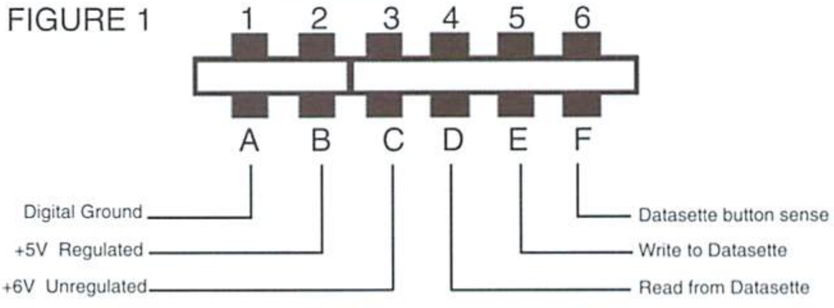
1. You can ignore the cassette port entirely and live a full and happy life using your disk drive.

2. You can use the cassette port solely as a source of regulated power for some external device. This kind of usage was popular with some early user-port hookups. Early PET and CBM systems didn't have a regulated voltage source on the user port, so an extra connector often tapped power from the cassette port next door.



3. You can use the port as a source of switched unregulated voltage. There's enough power there to run the motor of a Datasette, so with a little finagling, you can arrange a handy medium-power output.

FIGURE 1



The Cassette port edge connector. Upper and lower pins are shortened. Note that this connector is keyed between pins 2 and 3.

4. You can use the port for general data input and output. Those input and output lines are generally usable. The input line, in particular, is configured nicely to be an interrupt source. That's how it works in regular Datasette service.

5. You can use the port, with its supporting ROM logic, as a means of passing data or programs to a compatible Commodore machine. In this area, the Commodore 16 and Plus-4 are not compatible with the others, since they use their own tape format. Apart from this, a null-modem style cable can be a very effective way of moving data, although it's not super fast. The nice thing is: the protocols for saving/loading or data writing/reading are already in place in the ROMs.

6. Finally, you can use the port as Commodore intended, that is, to read and write data with cassette tape. Commodore designed the Datasette specifically for this job. Hackers with some mechanical and electrical talent can adapt a conventional audio tape recorder to do the same job. The tricky part is automatic starting and stopping of the tape motor, but you can sometimes work around this.

ROM Routines

If your computer has been fitted with a customized ROM, you can't be sure that the cassette port logic is still in place. Many third-party systems do marvelous things to help the user and programmer. But the new features have to find room inside limited ROM space, and usually something has to go to make room. The casualty is most often all or part of the cassette port routines.

If you have a custom ROM system and want to do something using the cassette port, it would be a good idea to check to see what still works. Some ROM revisions, such as JiffyDOS, come with a switch so that you can restore the old ROM if you should need it. Then again, if your objectives are largely hardware in nature, you may not need to worry about whether the original ROM is still in place.

The Hardware Setup

Figure 1 shows the connections to the edge connector. Here's a more detailed description about where these connections go:

The 5 volt regulated power (pin B) hooks up to the computer's main supply. The switched power on pin C isn't completely unregulated. The raw supply of about 9 volts is switched through a power transistor. The voltage level is typically guarded by user of a zener diode, which limits it in most cases to the 6 or 7 volt range. Actual details may vary, especially in countries outside North America.

What switches this power, originally intended for the Datasette motor? It's done via an I/O port. On the Commodore 64 and 128, it's the 6510 on-chip register that is found at addresses 0 and 1 (bit 5). Watch it: the interrupt plays with this information, too. (See the note on this later.)

Output to the Datasette is sent on pin 5, and it's hooked up just about the way you would think, directly from the I/O chip. On the 64/128, that would be bit 3 of the same on-chip register. Continuing the bit lineup: bit 4 is used to detect whether the Datasette is asking for power. If any "locking" key such as play, forward, or rewind is down, this signal will be activated on pin 6.

The remaining input signal, data input, has some tricky aspects. This one is delivered to a completely different I/O chip—on the 64/128, it's the CIA "flag" bit, located in the ICR, the interrupt control register. That's located at address 56333, hexadecimal DC0D, bit 4. Although you could test this bit directly, this register is normally used to trigger interrupts, and the critical timings involved in tape input make this the best way. This is especially true because the "flag" bit is edge-triggered, which means the program has to reset it after each signal.

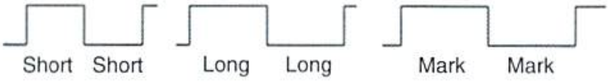
Tape Motor Interlock

When the Datasette buttons are up, the computer never supplies power to the motor. When a locking button is down, the computer might supply motor power, and it might not.

The typical situation is that you get power when you ask for it (the interrupt routines do this). After a load or data read has been accomplished, the processor shuts off the cassette motor, even though the buttons may still be down.

If you want to control that unregulated power switch yourself, arrange for the hardware to signal a permanent "button down" condition. If you leave it as "button up", the IRQ interrupt will shut off the motor the moment your program tries to start it.

Figure 2a. The three signal elements of cassette tape.



Timings	Half-cycle	Frequency
Short	182 us	2.75Khz
Long	262 us	1.91Khz
Mark	342 us	1.46Khz

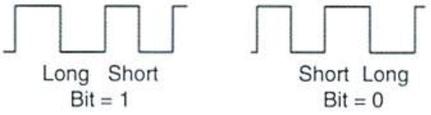


Figure 2b. Data bits on tape are made up of a long and a short cycle. The order determines whether it's a 1 or a 0 bit.

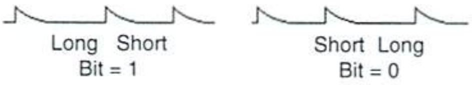


Figure 2c. When reading tape, only the rising edge of the sign triggers the interrupt. This makes signal polarity important.

Tape Signals

Figure 2 shows details of the kind of signals that go to tape. In figure 2a, you see that three basic frequencies are sent: a "short" signal, a "long" signal, and an even slower "mark" signal that is used to indicate the start of a data byte.

The frequencies involved go outside the normal range of telephone circuits. The telephone company gets upset when you try to send signals higher than about 2300 cycles per second. So the cassette interface won't work as a substitute modem.

At the start of any tape block, there's a "leader", a sequence of high-pitched "shorts". By the way, avoid playing such tapes on your home audio recorder because the binary signals tend to cause head magnetization, which will spoil the quality of other listening.

Then a "long-mark" sequence indicates the start of a byte, followed by either a long-short combination for a "1" bit, or a short-long combination for a "0" bit. A ninth bit is added to each byte to generate odd parity for error control. When the computer reads these incoming signals, it uses only the "leading edge". Figure 2c reflects this. A little thought will reveal that the signal polarity is important; turn the signal upside down, and the timing becomes unusable.

Standard Format

A program written with SAVE will consist of two chunks written to tape. The first chunk is a header which is always 192 bytes in length. The second chunk is the program data, which may be any length.

A data file is written by means of the conventional OPEN..PRINT#..CLOSE sequence. It consists of two or more chunks, each of which is 192 bytes in length. Again, the first chunk is a header. The remaining chunks contain data. The unused part of the last chunk is filled with binary zeros. This signals end-of-file, so you can't write binary zeros into a cassette data file.

Whether the file is a program or data, each chunk is written twice. This allows for error correction during reading.

Snooping Headers

It can be interesting or useful to examine header blocks written on tape. The header block will be found in the cassette buffer. On the 64, this is located at addresses 828 to 1019 (hex 033C to 03FB). The contents of the first byte, as seen with PRINT PEEK(828), will be 1 or 3 for a program header, 4 for a data file header, 2 for a block containing data, and 5 for an "end-of-tape" block. There's usually other information in the block, such as the file name. I'll leave this one for your experimentation.

You can force the Commodore 64 to perform a single header-block read by commanding SYS 63553. The corresponding 128 command is SYS 59890; look for the 128's cassette buffer starting at address 2816 decimal.

There's a lot more to be said about the cassette system. The information given so far will be a good start for curious users.

ASSEMBLY LINE

ORGANIZATION AND INPUT

Jim Butterfield

We'll talk briefly about the three parts of a machine language program, and how these parts are usually set up in memory. Then it's on to an exercise in input, using the GET subroutine at hexadecimal FFE4.

Code, Constants, and Variables

A typical program is made up of three parts. The "code" section contains the machine language instructions, so we often think of this as the whole program, but there's more.

The "constants" part of the program is where we put all other fixed data, like strings to be printed, numeric values, and even "pointers". Pointers are two-byte locations containing addresses that point at locations in memory. They often point at parts of your program.

The "variables" part holds data items that change as the program runs. Again, these might be anything; strings, values, or pointers. The program sets up initial values for variables and changes them as it runs. Figure 1 shows the three parts of a program as it might be set up in memory.

A program, as it is saved on disk, needs to contain the "code" and "constants" sections. The "variables" will be set up during the program run. We save the program as a single chunk. That means it's a good idea to keep the code and constants sections close together.

You can leave gaps between the various elements: your main program, its subroutines, and various types of constants. Figure 2 illustrates this. If you're using a tiny assembler, you'll find it useful to allow such gaps, since you can't be sure what part of memory will be available as you are writing the code. Keep each code module—main routine or subroutine—compact. "Branch" instructions are limited to a reach of approximately 128 bytes. You don't want to put gaps within a module that might put your branches out of reach.

Variables

Beginning programmers sometimes have problems with the concept of variables. It's often useful to think of them as "blanks to be filled in". They might be filled in by the user who types data, or they might be filled in by the program, which uses variables for such things as counters, totals, or buffers for file input or output. Don't forget that many variables need to be initialized before use. It's not like BASIC, where all variables are known to be zero when the program starts.

In our project, we will use two variables. The first is a byte where we can save an index value. The second is a buffer to hold an input string typed by the user. Typically, a string wouldn't be longer than 80 characters, but it could exceed that on some models of Commodore computer.

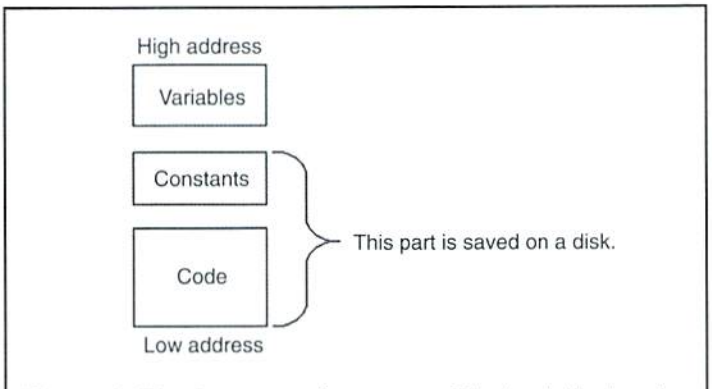


Figure 1. The three parts of a program. The "variables" part doesn't need to be saved on disk, since values will be set up as the program runs.

Program Project

Here's our project: invite the user to type in a string, and then repeat back that string. The computer will need to store all the characters received, so we'll put them in an area called a "buffer".

The coding shown here is appropriate for most tiny assemblers, as would be found in a Machine Language Monitor. Of course if you happen to have a symbolic assembler, you can use that. To get input from the keyboard, we will call the GET subroutine at address $FFE4. This subroutine reads from the keyboard buffer. The program will need to "echo" the received keys so that the user can see what has been typed.

Here's a warning: When you call this routine, you cannot be sure that registers X, Y, and A will keep their previous contents. Register A will contain the incoming character (or binary 0 if no character is ready), so we expect it to change. We also cannot expect the contents of X or Y to be preserved. If you have an important value in any register, save it and bring it back after the call has been completed. Only the CHROUT routine at $FFD2 reliably preserves all registers. I'll write the program to start at address hex 2000, decimal 8192.

The Program

Don't type this in until you read the comment below giving values to substitute for XSAVE and BUFF.

```
A 2ØØØ LDX #$ØØ      (start of buffer index)
```

Here's where we loop to get more input.

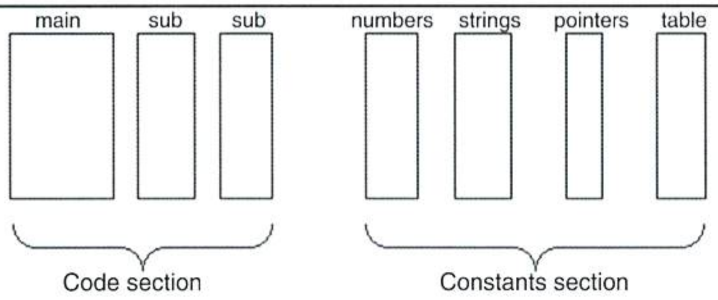


Figure 2. You may leave gaps between the various parts of the program; in fact, the elements can go in any order. However, if the gaps are too big, you'll waste disk space and lengthen loading time.

```
A 2ØØ2 STX XSAVE    (store x...  )
A 2ØØ5 JSR $FFE4    (.. get character ..)
A 2ØØ8 LDX XSAVE    (  ..restore X)
```

The next line is a quick way to test if A contains a zero:

```
A 2ØØB TAY          (binary zero?)
A 2ØØC BEQ $2ØØ2    (if yes, try again)
A 2ØØE STA BUFF,X   (store character in buffer)
A 2Ø11 JSR $FFD2    (and echo it to the screen)
A 2Ø14 INX          (add to the buffer index)
A 2Ø15 CMP #$ØD     (did we just do a RETURN?)
A 2Ø17 BNE $2ØØ2    (if not, back to the loop)
```

Since we've just received the RETURN character, our input line is complete. That character has been echoed and stored like all the others.

```
A 2Ø18 LDX #$ØØ     (reset buffer index)
```

Here's where we loop to do output.

```
A 2Ø1B LDA BUFF,X   (get a character)
A 2Ø1E JSR $FFD2    (print it)
A 2Ø21 INX          (move buffer index along)
A 2Ø22 CMP #$ØD     (have we just done a RETURN?)
A 2Ø24 BNE $2Ø1B    (if not, loop to do more)
A 2Ø26 RTS          (if RETURN, quit)
```

Cautionary Notes

Be sure you don't forget the "#" characters where they are needed. There's a lot of difference between loading the value zero, and the contents of address zero.

To read over your code, remember that you can disassemble using the D command. Type D 2000, and you'll see a screen full of disassembled code. This allows you to check for accuracy. If you see a mistake, you can type over it on the right-hand side of the display. To see more, just type another D with no address.

Remember that you can put in NOP (no operation) commands at "interesting" locations. Later, you could change them to BRK, the "break" command, so as to stop the program and see what has been happening up to that point. Your program will be slightly bigger, but you'll be able to test it as it runs.

No Constants, Two Variables

The above program needs to use two variables: a single byte called XSAVE, and a buffer area whose start we have called BUFF. We could place these directly behind the code, putting XSAVE at address $2027 and BUFF at address $2028 (going up to, say, $2078). A symbolic assembler would usually put them there, so we'd ask it to give us two variables with code like:

```
XSAVE  *=*+1
BUFF   *=X+8Ø        (most assemblers support decimal values)
```

If you're chopping away with a tiny assembler such as you get with a machine language monitor, it's easy to leave some space and pick a higher location for these variables. After all, we won't save variables as part of the disk program, so the "waste space" won't hurt. I'll pick addresses $2100 and $2101 (and up) for the two variables.

So, substitute $2100 for XSAVE wherever it appears above, and substitute $2101 for BUFF. Enter the program, return to Basic, type SYS 8192. The screen will be quiet. then type in a line followed by RETURN and the program should echo it.

Expanding the Program

We won't try to fix that missing cursor this time, but you might like to add code so as to print a prompt message, such as "TYPE SOMETHING:" before accepting input, and another string such as "I GOT:" before echoing what was received. These two "canned" strings will be constants, and they should be placed behind your program. Address hex 2050 for the first string should give you plenty of room for your program. Don't forget that you put in these strings with the M (memory display) command. They are not instructions, so you do not use "A for assemble".

I'll leave this as an exercise, but here are a few hints. If you followed my suggestion to put the print string at about $2050, your code would start out along the following lines:

```
 LDX #$ØØ
($2ØØ2) LDA $2Ø5Ø,X
 JSR $FFD2
 INX
 CMP #$1Ø
 BNE $2ØØ2
```

I'm assuming a string of length 16 bytes, which is where the #$10 comes from above. Adjust this value to suit your own string.

This should take you from $2000 to $200C, so the next segment of code will start at $200d. That's the code to actually input the string. You can copy it from the earlier program above, but keep in mind that the loop will NOT go back to $2002, so adjust the BNE address to a suitable value. That code will go from about $200D to $2025. Next is the constant string for the echo (my suggestion was "I SAW: "). This will be similar to the code for the previous constant string, above, with a change of addresses. This part might fit in memory from $2026 to $2032. Finally, the echoing of the string will be done with code at about $2033 to $203F, with the final RTS at $2040. Your code might be a little longer. This leaves about fifteen empty bytes before our string constants at $2050, but that's not much wastage. If we were to save this program, we'd save the whole thing from $2000 to $2067 or so, code and constants, but not the variable area.

CHECKSUM

Commodore World's Program Entry Checking Program and Tips on Entering Programs from this Magazine

CHECKSUM is a program that proofreads your typing when you enter a listing from the magazine. It assigns a numerical value to each character that you type, adds up the values of the line you typed and displays the sum. (Checksum, therefore, means that it checks your typing by summing the characters.) It also verifies that you have typed the characters in the proper order. (Checksum won't tell you if you miss a line of code entirely, so verify that yourself.) Checksum runs "in the background" when you type in lines of program code. Whenever you type a line and press RETURN, Checksum will display a value. Compare that value to the value published next to the line of code in the magazine. If the numbers match, you've typed the line correctly. Simple.

Typing in CHECKSUM

First, type in Checksum carefully from the listing on this page. Be sure to press RETURN after every line to enter it into memory. Once you have typed the program, *save it*. In fact, save it a few times while you're typing, just to be safe. (This is good advice whenever you type in a program. I usually change the name each time I save; for example, Checksuml, Checksum2, and so on.) Double-check your work, making sure that you've typed in every line and that you've pressed RETURN after every line you've typed. If you make errors when typing in Checksum, a test run of Checksum will tell you which line is incorrect. (This sarety feature works only in the Checksum program itself, and does not apply to any other listings in the magazine.) Whenever you find a typing error (in any program listing), fix it, press RETURN to enter the change, save the program again and try another run. Repeat this process as often as necessary. Important tip: Don't get discouraged if the program won't run. Be patient. Be thorough. It will work eventually. You'll know your Checksum is ready when you see the line:

```
TO TOGGLE ON OR OFF, SYS XXXX
```

Entering Programs Using CHECKSUM

When you're ready to type in your first listing from the magazine, load and run Checksum. Make a note of the number that is displayed on the screen (49152 for the C-64; 3328 for the C-128). To activate and deactivate Checksum, type SYS followed by that number, then press RETURN. You need to have Checksum active whenever you're typing in a listing. Checksum must be deactivated, however, when you run the new program. The next step is typing in a new program listing as it appears in the magazine.

As you begin, you'll notice that to the left of the start of each line is a number. Don't type this number in: It's simply the Checksum value. Stop typing at the end of the program line and press RETURN. If you've typed the line correctly, the number displayed on the screen will match the Checksum value. If the numbers don't match, you've made a mistake. Check the line carefully, make your changes and press RETURN. The computer won't know you've made a change unless you press RETURN on the changed line to enter it. A few type-in hints: The Checksum does not verify blank spaces in the program lines unless they are within quotation marks, because adding or omitting such spaces will not affect the operation of the program. The exception to this is hexadecimal Data statements. These are the Data statements, such as this one, that don't have commas:

```
100 DATA 12345678901234567890*123456789012345
67890*12345678901234567890*
```

In statements such as these, you must have one space between the word DATA and the numbers that follow. Checksum will not catch that error.

Special Key Combinations

As you type, you may be confiused the first time you see curly braces {}. These braces mean "perform the function explained within." For example, {22 SPACES} means that you need to press the space bar 22 times. Don't type the braces (you can't, of course, because there are no curly braces in the Commodore character set). Here are some other common examples:

{CLEAR/HOME}	hold down the SHIFT key and press the CLR-HOME key.
{2 CRSR DN}	tap the cursor down key twice.
{CTRL i}	hold the CONTOL key and press the I key.
{CMDR t)	hold down the COMMODORE key and press the T key.

Continue typing in your program, saving often and checking each checksum value with the one in the magazine, until you've finished the listing. Phew! So now you're ready to run your program, right? Not quite. First, save it. Second, deactivate Checksum by typing SYS followed by 49152 for the C-64 or 3328 for the C-128. Now you can run. Don't be discouraged if you still get an error. It happens. Use Checksum faithfully. Be patient. Be thorough. It will work eventually.

CHECKSUM

```
100 rem cw checksum 64/128
11Ø mo=128:sa=3328
12Ø if peek(65533)<>255 then mo=64:sa=49152
13Ø i=Ø:ck=Ø:ch=Ø:ln=3ØØ
14Ø for k=Ø to 16
15Ø for j=1 to 1Ø
16Ø read b:if b>255 then goto 28Ø
17Ø ch=ch+b:poke sa+i,b:i=i+1
18Ø next j
19Ø read lc:if lc<>ch then goto 28Ø
2ØØ ch=Ø:ln=ln+1Ø
21Ø next k
22Ø pokesa+11Ø,24Ø:pokesa+111,38:pokesa+14Ø,234
23Ø printchr$(147):print"cw checksum";str$(mo):print
24Ø print"to toggle on or off, sys";sa:if mo=128 then 27Ø
25Ø pokesa+13,124:pokesa+15,165:pokesa+25,124:pokesa+26,189
26Ø pokesa+39,2Ø:pokesa+41,21:pokesa+123,2Ø5:pokesa+124,189
27Ø pokesa+4,int(sa/256):sys sa:new
28Ø print"you have a data error in line";ln;"!":end
29Ø rem do not change these data statements!
3ØØ data 12Ø,162,24,16Ø,13,173,4,3,2Ø1,24,884
31Ø data 2Ø8,4,162,13,16Ø,67,142,4,3,14Ø,9Ø3
32Ø data 5,3,88,96,32,13,67,152,72,169,697
33Ø data Ø,141,Ø,255,133,176,133,18Ø,166,22,12Ø6
34Ø data 164,23,134,167,132,168,17Ø,189,Ø,2,1149
35Ø data 24Ø,58,2Ø1,48,144,7,2Ø1,58,176,3,1136
36Ø data 232,2Ø8,24Ø,189,Ø,2,24Ø,42,2Ø1,32,1386
37Ø data 2Ø8,4,164,18Ø,24Ø,31,2Ø1,34,2Ø8,6,1276
38Ø data 165,18Ø,73,1,133,18Ø,23Ø,176,164,176,1478
39Ø data 165,167,24,125,Ø,2,133,167,165,168,1116
4ØØ data 1Ø5,Ø,133,168,136,2Ø8,239,232,2Ø8,2Ø9,1638
41Ø data 169,42,32,21Ø,255,165,167,69,168,17Ø,1447
42Ø data 169,Ø,32,5Ø,142,169,32,32,21Ø,255,1Ø91
43Ø data 32,21Ø,255,169,13,32,21Ø,255,1Ø4,168,1448
44Ø data 96,1Ø4,17Ø,24,32,24Ø,255,1Ø4,168,96,1289
45Ø data 56,32,24Ø,255,138,72,152,72,24,162,12Ø3
46Ø data Ø,16Ø,Ø,32,24Ø,255,169,18,2Ø8,198,128Ø
```

Refurbished Units & Repair Service !

C-64 Refurb. with P/S (No books)	$ 54.95
C-64 Refurb. without P/S (No books)	46.95
1541 Refurb. with cord, serial cable (No books/disks)	64.95
Monitor Cable, 5-pin DIN to RCA jacks	4.95
Serial Cable, 6-pin DIN, 6'	8.95
Commodore Power Supply for C-64 (used, tested)	7.95
Power Cord for 1541	3.95
Power Supply for 1541-II	19.95

Printers/Monitors: We have a limited supply of C-64 compatible printers and monitors. Write for current list & prices.

Ribbons: We stock ribbons for most Commodore and C-64 compatible printers. Write for current list & prices.

Service: We will repair or replace:

C-64 (not including P/S)	$ 35.00
1541	$ 45.00

Service price includes parts and labor. Send computer or disk drive with name, address, phone #, & describe problem. Repair or refurbished replacement will be shipped UPS.
(Please remove any non-Commodore upgrades)

Commodore Repairs
Since 1979
• • • • • •
90 Day Warranty
on all repairs
and refurbished items.
• • • • • •
Write for
unadvertised specials!

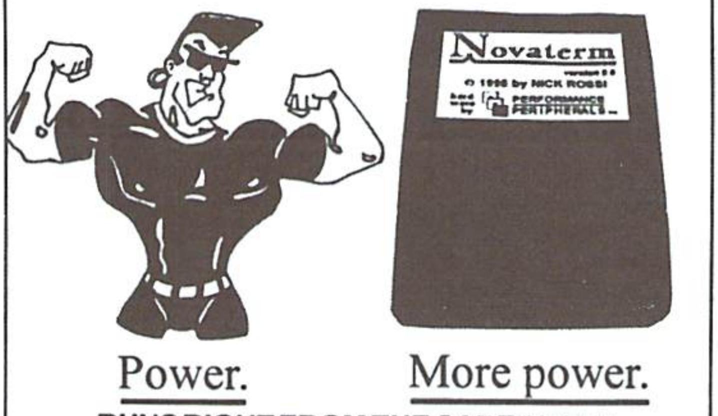
TO ORDER: Send check or money order for the total of your order plus $6.00 shipping (Personal checks take 2 weeks to clear). For COD orders, add $5.00. We currently do not accept credit cards.

Unique Services (413) 496-9275
68 Velma Avenue, Pittsfield, MA 01201
All items subject to availability. Prices subject to change.
MA residents add 5% sales tax (except for service)

Novaterm

version 9.6

Modem software for the Commodore 64



Power. More power.

RUNS RIGHT FROM THE CARTRIDGE!
PORT EXPANDER INCLUDED!

- **80 columns** with C64/C128
- **ANSI, VT102,** and C= graphics
- **Zmodem** upload and download
- **Script** language for automation
- **38,400** bps w/ CommPort or SwiftLink, **57,600** bps w/ HART
- **REU, RAMLink, RAMDrive, BBG/GEORam** as a big buffer

"I must admit, it's better than my new operating system." - Bill G., Mykro-soft Corp.

Bring the telecommunications revolution to your Commodore 64
TO ORDER: CALL (800) EASY-WEB Visa, MC, etc.

USED SOFTWARE

We buy, sell, and trade used original software. Lowest prices for C64, C128, Amiga, and IBM. Mention this ad and your computer type for a free list. Call our BBS for a complete list within minutes.

BBS:
8N1 1200-28.8k
616-429-7211

Ask for a list via Internet:
CENTSIBLE@DELPHI.COM

Call or write:
CENTSIBLE SOFTWARE
P.O. Box 930
St. Joseph, MI 49085
Phone: 616-428-9096

ESCAPE ROUTE

The Adventures of Eric Hawthorne, P.I.
A combination of action, board, and strategy games. Help Eric track the infamous ICELADY who has stolen the rare Acme Diamond! Fun for everyone! For the C-64 or C-128 in 64 mode.
Send check or M.O. for $19.95 to CREATIVE PIXELS, P.O. Box 592, Library, PA 15129

COMMODORE COUNTRY

EverythingforCommodoreComputers
Sell • Trade • Repair • Buy
1420 County Rd. 914
Burleson, TX 76028
817-295-7658
817-447-6974 - Voice/FAX line
ALL STORE ITEMS HAVE A 90 DAY WARRANTY. IF, FOR SOME REASON YOU ARE DISSATISFIED IN YOUR PRODUCT, YOU MAY RETURN IT FOR REPLACEMENT OR STORE CREDIT ONLY. SORRY, NO REFUNDS.

C64/1541 Repair - $40
SPECIAL
C64, 1541, J-S+Sft. - $99

We carry a full line of hardware, software & magazines, both new & used, including European items. Our flat-rate repairs include both parts & labor. Call for details. Trade in your unwanted items.
Catalog - $2.95
ALL MAJOR CREDIT CARDS ACCEPTED.
We carry a full line of Amiga Products.

All Aboard the Commodore Information Supersubway!

The Underground is a grass-roots Commodore publication produced with 8-bit computers, Geos, and a Postscript laser printer. For a mere $11 per year, you'll get helpful features, current reviews, and columns on everything from projects to PD software. A sample issue of the Underground is only $2--why not give it a try?

THE UNDERGROUND, 4574 Via Santa Maria, Santa Maria, CA 93455
Can/Mex: US$2.25/sample, $12.50/one year. International: US$3.75/sample, $21.50/one year.

* CLASSIFIED ADS *

C64/128 PUBLIC DOMAIN. REQUEST FREE CATALOG OR SEND $2 FOR A DEMO & CATALOG. CALOKE IND., P.O. BOX 18477, RAYTOWN, MO. 64133. VISA-M/CARD ACCEPTED.

C-64 FOREIGN-AMERICAN Utilities, Graphics, Hacker, Arcade. 32¢ stamp gets catalog. Home-Spun Software, POB 1064-CW,Estero,FL. 33928

CMD 4MB RAMLink, $250. C-64 games: AD&D all 3 Zorks, Jet + 3 scenery disks, Sons of Liberty, Defender of the Crown, $45. V. Peterson, RR2 Box 59, Thompson, IL 61285.

GEOS Publication. The exclusive monthly GEOS publication. Beginning Feb. 1st, rates will be $12.00 for one year; $20.00 two years.Write 713 E. Main Street, Independence, KS. 67301-3726.

Wanted - "Pirates" game disk for Commodore 64. Donald A. Wilson, 17005 Westdale, Cleveland, OH 44135. Will pay.

Wanted - New and mint used commercial games for Apple 2. No public domain. ETH, 2900 N. Apperson Way Lot #33, Kokomo, IN 46901.

CW & RUN BACK ISSUES

For years, *RUN* Magazine provided Commodore Users with a great source of information, and now CMD has given you *Commodore World*. Don't let this valuable information slip away—fill in the voids in your library now!

RUN Magazine Back Issues

Any 3 issues for $12.00, any 6 for $18.00, or any 12 for only $24.00

RMJAN88	January 1988	**RMOCT89**	October 1989
RMFEB88	February 1988	**RMSP89**	Special Issue 1989
RMMAR88	March 1988	**RMJAN90**	January 1990
RMAPR88	April 1988	**RMMAY90**	May 1990
RMMAY88	May 1988	**RMJUN90**	June/July 1990
RMJUN88	June 1988	**RMDEC90**	December 1990
RMJUL88	July 1988	**RMJAN91**	Jan/Feb 1991
RMFEB89	February 1989	**RMJAN92**	Jan/Feb 1992
RMMAY89	May 1989	**RMMAR92**	March/April 1992
RMJUN89	June 1989	**RMMAY92**	May/June 1992
RMJUL89	July 1989	**RMJUL92**	July/August 1992
RMAUG89	August 1989	**RMSEP92**	Sep/October 1992
RMSEP89	September 1989	**RMNOV92**	Nov/December 1992

Shipping: 3-6 issues - U.S. $3.00, Canada $5.00, Foreign $15.00; 12 issues U.S. $5.00; Canada $7.00; Foreign $20.00.

Commodore World Back Issues

$4.95 each, or any 3 for only $12.00

CW1	Issue 1, Volume 1, Number 1	**CW7**	Issue 7, Volume 2, Number 2
CW2	Issue 2, Volume 1, Number 2	**CW8**	Issue 8, Volume 2, Number 3
CW3	Issue 3, Volume 1, Number 3	**CW9**	Issue 9, Volume 2, Number 4
CW4	Issue 4, Volume 1, Number 4	**CW10**	Issue 10, Volume 2, Number 5
CW5	Issue 5, Volume 1, Number 5	**CW11**	Issue 11 Volume 2, Number 6
CW6	Issue 6, Volume 2, Number 1	**CW12**	Issue 12, Volume 2, Number 7

Shipping: U.S. and Canada $2.00 for first issue, plus $1.00 per additional issues.; Foreign $5.00 per issue.

TO ORDER CALL 1-800-638-3263

ADVERTISERS INDEX

COMMODORE WORLD
Classified Advertising

Commodore World subscribers may place non-commercial classified advertising in *Commodore World* at a cost of $10.00 per issue. Your advertisement may contain up to 150 characters (including spaces). Send your advertisement with payment to: CW Classified Advertising, c/o Creative Micro Designs, Inc., P.O. Box 646, East Longmeadow MA 01028-0646.

Don't wait until it's too late...

RENEW EARLY!

Is your Commodore World subscription close to running out? Here's an easy way to check: look at the mailing label on the front of your copy. You'll find your subscription number and the expiration issue number. For example:

```
James Smith  12345EXP16
123 Home Street
Grand Rapids, MI  49502-0123
```

Jim's subscription will run out with Issue 16 as indicated by the EXP14 in his subscription code. Jim would be wise to re-subscribe early to avoid missing a single issue of Commodore World!

COMMODORE WORLD

THE NEWS MAGAZINE FOR COMMODORE 64 & 128 USERS

COMMODORE WORLD

Issue 1
Volume 1, Number 1
April 15, 1994
$4.95 U.S.
$5.95 Canada

PREMIER ISSUE

THE NEWS MAGAZINE FOR COMMODORE 64 & 128 USERS

Printers Our in-depth look into selecting and using printers and printer interfaces

CMD DOS An introductory look at what makes CMD devices tick

GEOS Steve Vander Ark takes a stab at Morphing with GEOS

Review QWKRR 128 simplifies offline replies to FidoNet

Plus...
- On The Horizon
- Just For Starters
- Top Tips
- BASIC Instincts
- Advanced Techniques
- geoProgrammist
- Carrier Detect

STAY IN TOUCH!

CALL CMD

1-800-638-3263

To Subscribe

Annual Subscription
only $29.95
8 issues per year

Commodore World is the publication that will keep you informed in these times when up-to-date information on Commodore computing is so hard to find. Published by Creative Micro Designs, the industry leader in development of Commodore-related products for over six years, Commodore World will supply you with information on what's new, what's still available, and above all else—*where* to get it. If you felt you had nowhere to turn to for Commodore support, turn to the pages of Commodore World for a wealth of resources ready to help you get the most from your computer!

You'll find Commodore World feature articles informative and easy to read; what's more, they're written by leading authorities and experts, many of whom have written for other Commodore-related publications in the past. And Commodore World has something for everyone, whether you're a novice or an experienced programmer.

And while our feature articles help to cover different subjects in each issue, regular columns provide on-going insight into topics of interest to most users. You'll find columns that cover BASIC and advanced programming, and even a column for GEOS programming. And if you prefer being a GEOS user to being a GEOS programmer, you'll find another column devoted to helping you get more out of GEOS. If you want to learn more about using and programming the various peripherals on your system—you guessed it, we've got a column for that as well. Even first-time Commodore users will find a column devoted specifically to their needs.

And there's even more. Departments that cover news, telecommunications, reviews of available hardware and software; even news of what's happening in other Commodore-related publications!

So, if you really want to get the most from your Commodore, there's no better way to get it than Commodore World!

Columns

Just For Starters - An introduction to the C64/128 by Steve VanderArk

Foreign Exchange - An inside look at the market in Europe by Joseph Gaudl

Graphic Interpretation - GEOS, GEOS and more GEOS by Steve VanderArk

geoProgrammist - GEOS programming techniques by Maurice Randall

BASIC Instincts - BASIC tutorials and type-in programs by Gene Barker

Jim Butterfield's ML Column - Probably the best known name in our industry, Jim covers every aspect of programming in ML (coming soon)

Peripheral Vision - Technical insights to C-64/128 hardware peripherals

Carrier Detect - Exploring every facet of the Telecommunications experience

Over The Edge - Editorial covering various computer related topics and news

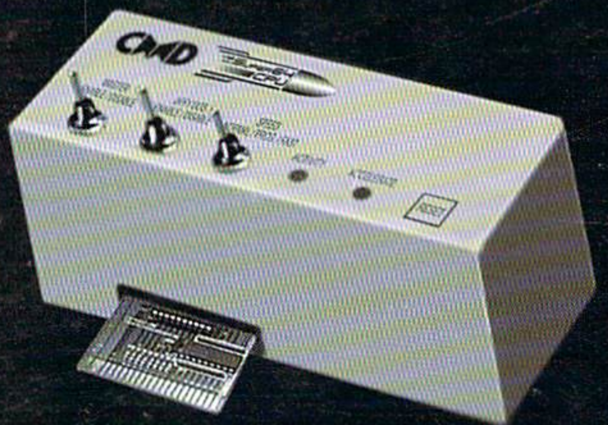
Departments

From The Editor • BackTalk • On The Horizon
Just Asking • The Connection • Top Tips
User Group Connection • Commodore Trivia
BBS Spotlight • Classified Ads

C-64 AND C-128 OWNERS!

SHOOT FOR THE STARS WITH CMD's

CMD SuperCPU's are designed to maintain an optimal level of speed and compatibility in a package that makes using it as simple as possible for the average Commodore user.



The SuperCPU is a cartridge port module that contains a new high speed microprocessor (65C816S) for your C-64 or C-128 computer, built-in JiffyDOS, pass-thru port and a compliment of switches for easy on-the-fly configuration. To the user, this means that your software runs as much as 20 times faster — that's right — 20 million cycles per second (cps) versus a stock C-64's 1 million cps. Not only does the SuperCPU speed up your computer's ability to crunch numbers, it will enhance screen updating, printing, disk access and any other function that utilizes your computer's processor. In short, the SuperCPU will help your computer bridge the gap between the Commodore and today's ultra-expensive computers, without sacrificing the many programs that are familiar to you!

To use the SuperCPU, just plug it into your C-64 or C-128 cartridge port and you're off like a rocket! Make your favorite software explode into the 21st century and prepare yourself for the exciting new titles that will appear in the coming months! Don't wait, place your advanced order today!

All specifications and prices subject to change without notice. Deposits will lock in at projected price.

Enclosed is my $50.00 deposit for CMD's all newSuperCPU!

Select: ❑ Super CPU 64 (20Mhz) ❑ Super CPU 128 (20Mhz)

Retail Price: $199.00 Retail Price: $299.00

Name ______________________________

Address ______________________________

City ______________ State/Prov. ______________

Country ______________ ZIP/Postal Code ______________

Home Phone ______________ Day Phone ______________

Payment: ❑ Credit Card ❑ Money Order ❑ Personal Check ❑ COD

Credit Card # ______________ Exp. Date ________ Bank Name ______________

We prosecute anyone placing fraudulent credit card orders

MAIL ORDERS TO:

Creative Micro Designs
P.O. Box 646
E. Longmeadow, MA 01028

PHONE ORDERS:

1-800-638-3263
(9:30 AM - 5:30 PM EST Mon. - Fri.)

AMERICAN EXPRESS Cards DISCOVER NOVUS MasterCard VISA