The Magazine For ALL Commodore Computer Users

TPUG

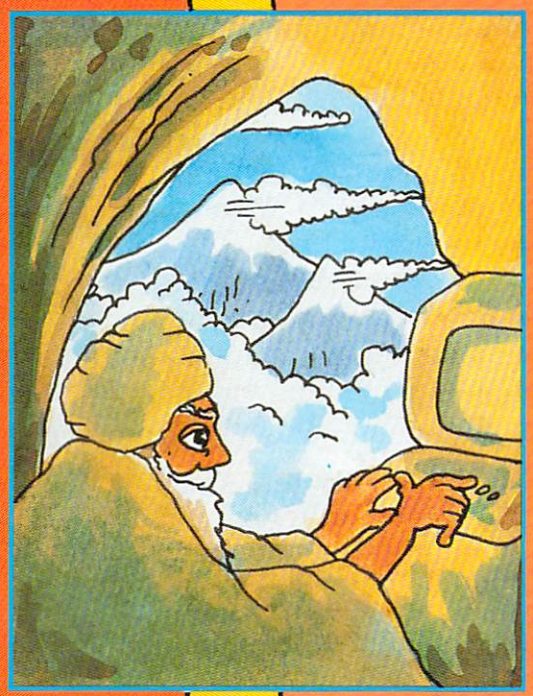
$2.95

Butterfield on C-128 Music

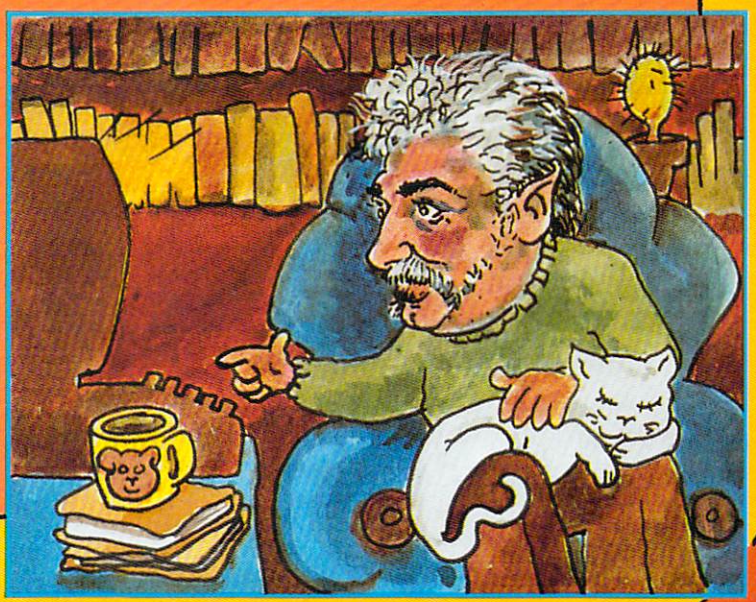
Punter on PUNTER

Forde: Wireless Computing

Fox: The Search Goes On



TELECOMMUNICATIONS



0 56698 72145 10

COMPLETELY MENU DRIVEN . VERY USER FRIENDLY
BETTER INTEGRATION THAN LOTUS 1-2-3 * OR SYMPHONY*

KING MICROWARE'S FULLY INTEGRATED

RHAPSODY 64

1 DATABASE MANAGER
2 WORD PROCESSING
3 SPELLING CHECKER
4 SPREADSHEET / GRAPHICS

NOW SHIPPING

RHAPSODY will allow you to have 18½k of file memory resident shared between text in the word processor, a spreadsheet, three dimensional bar charts and a data base, all instantly accessible by windowing to each menu.

All You'll Ever Need !

Suite 210, 5950 Côte des Neiges
Montréal, Québec H3S 1Z6
Tel.: (514) 737-9335

KING MICROWARE Ltd.

$64.95
DEALER INQUIRIES INVITED

*1-2-3 and Symphony are Trademarks of Lotus Development Corporation

TPUG Magazine

Publisher: Louise Redgers
Editor: Nick Sullivan
Assistant Editor: Marya Miller
Production : Astrid Kumas
Cover : Don Ballanger
Typesetting: Noesis, Toronto, Ontario
Printing: Delta Web Graphics, Scarborough, Ontario

TPUG Magazine is published 10 times a year by Toronto Pet Users' Group (TPUG) Inc., the world's largest Commodore users' group. TPUG is a non-profit corporation dedicated to the service and support of owners and users of Commodore computers.

Correspondence: Send change of address and subscription enquiries to: TPUG Inc., Address Changes, 101 Duncan Mill Road, Suite G7, Toronto ON, Canada M3B 1Z3. TPUG magazine welcomes freelance contributions on all aspects of Commodore computing. Contributions should be sent on disk, though accompanying hardcopy is welcome. Be sure to include return postage if you wish materials returned. Please indicate on the disk label which Commodore disk format and word processing program you have used. Remuneration for articles published is $30.00 per page if the author retains the copyright, and $40.00 per page if the copyright is assigned to TPUG Magazine. All contributions are subject to editing for length and readability. Address editorial contributions to: The Editor, TPUG Magazine, 101 Duncan Mill Road, Suite G7, Toronto ON, Canada M3B 1Z3.

Circulation:
Subscription 16,000 Newsstand 10,000
ISSN #0825-0367

Subscriptions to TPUG Magazine may be obtained by joining the Toronto PET Users' Group (TPUG) Inc. Another benefit of TPUG membership is the right to order inexpensive disks and tapes from the club's extensive software library. Yearly membership fees are:

Regular member (attends meetings)	$35.00 Cdn.
Student (full-time, attends meetings)	$25.00 Cdn.
Associate (Canada)	$25.00 Cdn.
Associate (U.S.)	$25.00 U.S.
	$30.00 Cdn.
Associate (Overseas – sea mail)	$35.00 U.S.
Associate (Overseas – air mail)	$45.00 U.S.

For further membership information please contact:

TPUG Inc., Membership Information
101 Duncan Mill Road
Suite G7
Toroto ON
Canada M3B 1Z3

1552 Hertel Ave.
Suite 144
Buffalo, NY
USA 14216-2882

TPUG Telephone Numbers:
Business Office (416)-445-4524
Magazine Office (416)-445-9865
Advertising (416)-445-4524

DIRECTORY

85 TPUG Magazine OCT

Inside Information

Rhapsody blooper

In the **Rhapsody 64** ad that appeared inside the front cover of last month's *TPUG Magazine*, the figure of 8 1/2K resident file memory was incorrect — by 10 K. The correct figure is 18 1/2 K of resident file memory.

TPUG Magazine regrets this error, and any inconvenience caused by it.

Mandrill mystery solved

The magazine office has been flooded with calls about the model with the Mel Gibson eyes on last month's cover. Many readers wanted to know who it was; some wanted to know *what* it was; and one wanted its telephone number. The remaining readers called to tell us that our headline: 'New Amiga Drives Competitors Ape' was inaccurate: our cover model was not an ape at all, but a mandrill, which is a monkey. Actually, we did know this, but we thought 'New Amiga Drives Competitors Monkey' didn't have quite the same ring to it — and 'Commodore Monkeys With Amiga' was right out altogether. Thanks anyway, all you simian purists.

This month

Telecommunications is probably the fastest-growing application for home computers. Though mass-market services like CompuServe have been around for several years, along with the more modest community-level single-user BBSs, it seems that telecommunicating is only now catching on in a big way. In this issue, you'll find articles explaining how to get acquainted with two of the biggest on-line services, CompuServe and Delphi, along with information of special interest to TPUG members using these services. We also have the first of two articles on the inner workings of the 'C1' file transfer protocol, by its author, Steve Punter; a rundown on telecommunications services offered by Bell Canada, by Ian Wright; a look at the public domain KERMIT protocol, by Phil Kemp; and lots more.

Our cover this month is by Toronto artist Don Ballanger. No prize will be awarded to the first ten readers who correctly identify the telecommunicator with the cat in the panel at the bottom right.

Amiga availability

As we go to press (October 9), it appears likely that Amigas will be available in Canadian stores within the next several days. The Computerland and Computer Innovations chains of computer stores will both carry the Amiga in Canada, as will the major independent dealers. Floor demos have already been sent out to 130 stores, and have been on display in some places for several days.

Early reports that the Amiga would be sold in two standard configurations are not correct, according to Roy Robinson of Commodore Canada. Instead, the computer will be sold in a basic package for $1999 Canadian ($1295 US), with additional hardware available separately. The basic package consists of the Amiga computer, with built-in 3 1/2-inch 800K disk drive, the already-renowned graphics and sound chips, and 256K user RAM; a detachable keyboard; a mouse; and four diskettes. The diskettes contain: 1) the Amiga's operating system, AmigaDOS, which must be loaded from the disk — called 'Kickstart' — every time you turn on the computer; 2) an 'iconic' interface, similar in spirit, though not in detail, to the user interfaces of the Macintosh and the Atari 520 ST; this is on a disk called 'Workbench'; 3) the BASIC interpreter, with an accompanying tutorial; and 4) two programs from Electronic Arts. One, called **Slideshow**, is a demo of their line of software for the Amiga; the other, **Polyscope**, shows off the Amiga's graphics capabilities.

The additional hardware for the Amiga should be available as soon as the basic package itself goes on sale. It will include an RGB monitor ($799 Cdn.), an external 3 1/2-inch disk drive ($450), and a 256K RAM cartridge that snaps onto the front panel of the Amiga ($300).

We are told that the current release of AmigaDOS (1.0) is much more reliable than prototype versions we had seen, and faster as well. AmigaDOS will nevertheless be provided on disk rather than in ROM for at least the next 12 months, to allow updates if further improvements are necessary. However, AmigaDOS loads into a special sealed-off 256K RAM area that does not reduce the minimum 256K available to the user.

Meanwhile, the C-128. . .

The C-128 is already available in stores in limited quantities. In Canada, a delay in getting CSA approval for the power supply means that each unit has to be individually inspected and approved by Ontario Hydro before it can be shipped — a pain for Commodore and for prospective customers. There is apparently a strong possibility that CSA will nix the external mounting of the fuse on the power supply, which is a pity. The 1571 disk drive should be available — again in limited quantities — within a week or so, according to one source inside Commodore Canada. The 1902 monitor, which allows both RGB and composite input, is not likely to surface in the near future. However, Zenith and Sanyo both manufacture RGB monitors that are known to work well with the C-128.

Delphi Sign-on

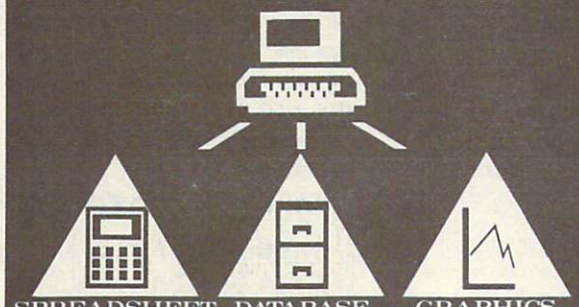
Here is the information you will need for signing on to the Delphi information service. As your sign-on username, enter: **TPUG**. And the initial sign-on password is: **ONLINE**. Have a good time!

Elections

Well, another year has passed and the elections for the Board of Directors have taken place. Congratulations to the new board members, and good luck with the challenges that lie ahead. Chris Bennett is our new President, Gord Campbell and Gerry Gold are the Vice Presidents, Carol Shevlin remains Treasurer and John Shepherd remains Secretary. A new office was created when Michael Bonnycastle decided not to run for President again. We wish to thank Michael for the years of dedication and energy that he has given to the club as its only President, and to welcome his participation as Past President. This stately turn of events suits him well.

The Editors

VIZASTAR 64
THE INFORMATION PROCESSOR



For COMMODORE 64

VIZA SOFTWARE

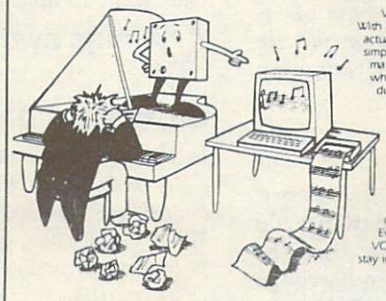
THE ULTIMATE HUMAN TO COMPUTER INTERFACE

VOICEMASTER

JUST IMAGINE

- YOUR COMPUTER TALKING TO YOU IN A REAL HUMAN VOICE. . . .
- YOUR COMPUTER UNDERSTANDING WHAT YOU SAY. . . .
- BEING ABLE TO WRITE MUSIC AND PLAY AN INSTRUMENT JUST BY HUMMING. . . .

THE VOICE MASTER - VOICE HARP



Voice Harp is a totally new musical concept. With Voice Master powerful software, you can actually compose and perform music in real time simply by humming, whistling or singing. It doesn't matter if you can't read music. Your voice or whistle pitch will write the notes for you including duration and rests.

As you hum or whistle, the notes scroll by on the video display. You can edit the notes, play them back, scroll them back and forth, deleting and correcting them. When finished, the score can be printed out.

In the performance mode, you can change octaves, musical keys or add chords. Even if you can't sing or have a sense of pitch, the VOICE MASTER will show you how easy it is to stay in tune.

HIGH QUALITY LOW PRICE

COMMODORE 64® SOFTWARE

The GOLD DISK

VOLUME 1 FEATURES:
- **Complete Data Base system** with:
  - — Full screen data entry
  - — Fast sorting capabilities
  - — Extensive multi-clause search facilities
  - — On line help facilities
  - — On disk tutorial and instructions
- **2 arcade style games** with 3D graphics
- **Learning to Program in BASIC:** Lesson 1 Input/output, variables and assignments, conditional branches.
- **Towers of Hanoi** with solution
- **The Personal House Purchase Analyzer:** Evaluation of financial scenarios

VOLUME 2 FEATURES:
- **Complete Word Processing Package** with:
  - — Cut-and-paste, full formatting capabilities. Margin Justification, Page Numbering
  - — Access to programmable printer capabilities
- **4 action screens game! "SPOOK CASTLE"!**
- **Children's educational game:** counting
- **Learning to Program in BASIC** — Lesson 2 Array & string variables: FOR-NEXT loop
- **8 Queen's puzzle** with solution
- **Loans and Mortgages program**
- **Files backup/copy program utility**

VOLUME 3 FEATURES:
- **A complete Typing Tutor program** Use the built in lessons or create your own! Complete with self evaluator
- **Arcade action game** "Chomper"
  - — 4 screens to complete at each level
- **Mind puzzling strategy game:** "Cubic"
- **Home Budgetter program**
- **Report Generator** (used with Vol. 1 database)
- **Tutorial:** Powerful sorting techniques
- **Learning to Program in Basic:** Lesson 3 PEEKs, POKES, print controls, multi-dimensional arrays and subroutines

VOLUME 4 FEATURES:
- **The Gold Disk Art package** High-res joystick controlled painter prgm with these powerful features:
  - — Preprogrammed shapes
  - — 'Fill' capabilities, Free drawing mode
  - — 'Undo' capability for error-correction
  - — 'Print' capability, Zoom
- **The Gold Disk Trivia Game** with:
  - — Over 500 questions, Multi player, scoring
- **Arcade action game** with 3D graphics
- **Educational:** File operations on the 1284
- **Programming Tutorial:** Binary Search technique illustrated
- **Complete directory program**

"I've always thought good programs cost a lot more. Now there's Gold Disk."
Karl Hildon, TRANSACTOR

Official approval from "Small Things Considered"
Radio WNYC AM83, New York

"A great way to get inexpensive quality software"
L. Redgers, TPUG Magazine

"The products low price, high quality and simplicity of use, make it an excellent addition..."
CLASSIC Bookshops

"Gold Disk remains on top." DASHER INDUSTRIES, Distributor

"...the right kind of product at the right prices."
COLES, THE BOOK PEOPLE

"Gold Disk idea's time has come."
Phil Kemp, INPUT Magazine

VOLUME 5 FEATURES:
- **The Gold Disk Sprite and Font Editor package** Joystick controlled graphic package for Sprites and character graphics with:
  - — Multi-colour capabilities
  - — High resolution capabilities
  - — Easy to follow instructions
- **Challenge game:** "Xode Breaker"
- **Arcade action, arcade graphics game:** "Cosmic Chaos" — Outstanding play!
- **Educational Tutorial:** How to Use Sprites
- **"Exights Two"** with solution

VOLUME 6 FEATURES
- **Enhanced Basic**
  - — Over fifty powerful new commands and functions
  - — Increases programmer productivity
  - — Simplifies programming
  - — Easy-to-use
- **Action Strategy Game** "Rogi Ginbani"
- **4 Screen Game:** "Return to Spook Castle"
- **Sound Tutorial** — Part I
- **Tutorial on** "Disk Manipulation"
- **Educational:** Encoding — Decoding
- **Financial Options Program** — Evaluate different payment scenarios

VOLUME 7 FEATURES:
- **Full Spreadsheet Program**
  - — Ability to evaluate "What if" scenarios
  - — Powerful and easy-to-use tool with
  - — Large Worksheet, Variable width columns
  - — Internal File Operations, Printer Option
- **Casino Pack** "Poker" and "Black Jack"
- **Bonus Card Game** "Solitaire"
- **Sound Tutorial** — Part II
- **Tutorial** on error recovery
- **Educational:** Stacks
- **Business:** Balance Sheet

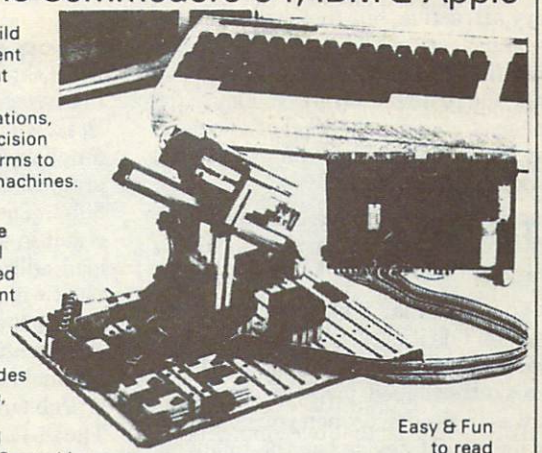
VOLUME 8 FEATURES:
- **FORTH!**
  The ease of BASIC, the Power of Machine Code
  Complete implementation of FIG FORTH!
- **The Gold Disk Trivia Game II**
  - — Over 500 new and interesting questions
  - — Multi-player, auto-scoring
- **Arcade action game** "Cube Slinker"
  3D action, multi-level
- **Programming with Forth** Part I
- **Tutorial** — Screen manipulation
- **Educational** — working with queues
- **Business:** Compound and Simple Interest

Fischertechnik®

Robotics Automat Kit

For the Commodore 64, IBM & Apple

Easily build 10 different intelligent robotic configurations, from precision robotic arms to sorting machines.



Simulate industrial automated equipment

Kit includes interface, industry standard Robotic Control Language based on Superforth

Easy & Fun to read user manual

QUALITY PRODUCTS AVAILABLE FROM

PHASE 4 DISTRIBUTORS

(403) 252-0911

FOR YOUR NEAREST DEALER CALL

1-800-661-8358

The Answer Desk

with Malcolm O'Brien

Restoring misaligned files

I realize that my 1541 disk drive gets out of alignment and requires correction. But is there no simple way to copy my misaligned disks to proper alignment? Or must I lose all my earlier efforts on a misaligned disk drive?

Gilbert R. Thornton
Longview, Texas

You don't have to lose your files, Gilbert, but I don't think that the solution could be called simple. I suppose that the simplest solution would be to borrow a second drive (an aligned one!) from a friend, attach it to your system, change the device address to 9 and then copy your files (with Jim Butterfield's **Copy-All** program, for instance) from device 8 (the sick drive) to device 9 (the healthy one). If a second drive is not available, you could use a datasette. A third possibility would be to upload your files to a BBS, align the drive, and download them again.

I realize that none of these alternatives is very attractive, but they will probably give you a strong incentive to decide which files really are expendable. Of course, the real solution is to minimize the chances (and consequences) of misalignment. There are several things that you can do in this regard:

- Ensure that your drive is situated in a place that is free from jarring shocks or vibration.
- If you find that you must move the drive, don't do it until you've put the head vibration protector into the slot. This is the diskette-shaped piece of cardboard that was in the drive when you bought it. This important device has the following words printed on it: 'It must be inserted and closed the door in transportation.' It's not good English, but it *is* good advice!
- As much as possible, avoid knocking the head against the drive stop. This is almost certainly the prime cause of misalignment. If you use software that knocks the head, try running the **drive saver** program beforehand. It's quite effective in a lot of cases.
- Head off misalignment before it becomes severe by running an align test program on the drive periodically. It only takes a few moments, and it's time well spent, if it saves you the headache of having to transfer your files as described above.

If problems persist, it would be worth your while to invest in an alignment kit, or to spend the money for a permanent fix.

Butterfield book title

In the Line Noise *section of your June/July issue, E.M. Hartston refers to 'Jim Butterfield's book on machine language'. What is the title of this book and who publishes it?*

G.S. Wattley
Pointe-a-Pierre,
Trinidad, West Indies

The book's title is *Machine Language For The Commodore 64 and other Commodore Computers*. It's published by the Brady Communications Company, Bowie, MD 20715. This publisher is a Prentice-Hall company. The book is probably available down there in the islands. If you can't find it, perhaps your computer dealer can order it for you.

B-128 transplant fix

In last month's instalment of *The Answer Desk*, we printed a short utility program for changing the load address of a B-128 program so that it can be loaded into an 8032. The program, as published, will result in a program file with the correct load address for the 8032 ($0401). The 8032 will correct the program line links during the load. This would be fine if we were converting a program from most Commodore machines; however, there is a problem when it comes to the B-128. The B-128 can legitimately have zeroes in the high bytes of the line link pointers. When the 8032 reads this zero it will suppose that it has reached the end of the file, and stop relinking. The result of all this is that you may end up with only the first line of your program (regardless of its length). Thanks to Jim Butterfield for pointing this out.

The bottom line of all this is that our conversion program must 'filter out' the zero bytes. The program necessarily becomes a bit longer and more complex. The conversion will take a bit longer too. Here's the amended program:

```
1 open 2,8,0,"infile"
2 open 3,8,3,"outfile,p,
  w"
3 get#2,a$,b$
4 c0=0: c2=2: c3=3: z$=
  chr$(0): l$=chr$(1)
5 print#c3,l$;chr$(4);:
  <space>goto 8
6 get#c2,a$: a=asc(a$+z$
  ): ss=st
7 print#c3,chr$(a);: if
  <space>a then on c2+(s
  s=c0) goto 6,10
8 get#c2,a$,b$,c$,d$: if
  <space>st or (a$=""and
  b$="") goto 10
9 print#c3,l$;l$;chr$(
  asc(c$+z$));chr$(asc(d
  $+z$));: goto 6
10 print#c3,z$;z$;:
   close 3: close 2
```

PC-compatible

A friend of mine told me that Commodore is selling an IBM-compatible computer in Canada. Why aren't they selling it down here in the States, and how compatible is it?

Elvin Glubang
Port Arthur, Texas

There are two, actually, Elvin — the PC-10 and the PC-20 (the PC-20 includes a hard disk). According to Commodore, all the computers that they sell in the United States are currently sold through mass merchandisers like K-Mart or Toys-R-Us. In Canada, the systems are being sold through computer dealers who can offer the level of dealer support that business users require. Although the situation State-side may change, Commodore currently has no plans to market the PCs south of the border.

On the bright side, the Amiga may well be in the stores by the time you read this. Word has it that there will be a PC emulator for the Amiga available on disk. But you'll probably need a 5 1/4 inch disk drive to take advantage of MS-DOS software (although some may be available in the 3 1/2 inch format used by the Amiga).

As far as compatibility is concerned, a local dealer here in Toronto has informed me that the PC-10 can be booted from a PC-DOS disk, and that it will run *all* PC software including 'tricky' ones like *Flight Simulator II* and *Sidekick*. □

Save Money On Canada's Most Popular Software By Mail.

ACTIVISION
The Next Generation.

Hacker (disk)	$34.95
Great American Road Race(disk)	$34.95
Countdown To Shutdown (disk)	$34.95
Gamestar Football	$32.50
Music Studio	$34.95
Toy Bizarre	$19.95
Zenji	$19.95
Hero	$19.95
Beamrider	$19.95

Hayes

SM 1200
1200 baud $599.95
SM 300
300 baud $239.95
Micromodem IIE
with SCI $249.95

Sega
C64
Spyhunter
$39.95

Broderbund

Printshop (disk) $49.95
Printshop Graphics II $29.95
Karateka (disk) $39.95
Lode Runner $29.95
Spare Change. $14.95

Simon and Schuster
C64
Typing Tutor III
$44.95

Newsroom
Newsroom Clip Art Collection
C64
$59.95

SPRINGBOARD™

Cardco
G Whiz Printer Interface for the Commodore C64 . . . . . $68.95
NEW S'MORE Memory Expansion . . . $70.00
(Adds 57% more memory)

Access
C64
Beachead II
$39.95
Mach 5 Cartridge
$34.95

Mindscape
C64
Halley Project
Crossword Magic
$39.95

Joysticks

Name Brand Joystick	$12.95
Trax-Ball	$9.95
Kraft Joystick — ATARI/C64	$14.95
Kraft — IBM	$59.95
Kraft — APPLE	$59.95

Screen Play
C64
Trivia Arcade
$14.95

Cosmi
C64
Super Huey
$24.95

Accolade
C64
Dambusters
$34.95

WINDHAM CLASSICS
C64
Treasure Island
Wizard Of Oz
Alice In Wonderland
$26.95

EPYX COMPUTER SOFTWARE
C64

Fastload (cart.)	$34.95
Summer Games II	$39.95
Winter Games	$39.95

Strategic Simulations
C64
Gemstone Warrior
Phantasie
$39.95

ELECTRONIC ARTS®
SKYFOX.
Fly the unfriendly skies.
Adventure Construction Set
Racing Destruction Set
$34.95

Better Working
C64
B.W. Word Processor
B.W. Spreadsheet
$39.95

FREE OFFER! Mail this coupon TODAY and receive your FREE SOFT MAIL catalogue.

Call our Canada-wide Toll Free Hot Line for Information and to Order.

1-800-387-3672

In Toronto: (416) 665-7323

Note: With orders, Ontario residents add 7% sales tax.
Add $2.50 for shipping.
Personal cheques welcome.

VISA MasterCard

Name: _____
Address: _____
City: _____ Prov: _____
Postal Code: _____ Tel: _____
Type of Computer: _____

soft-mail®

1085 Steeles Avenue West, Suite 405, Willowdale, Ontario M2R 2T1

Getting Started On Delphi

by John Easton

In the world of home computing, perhaps the last great unexplored territory lies in that vast world of telecommunications just a phone call away from your computer. *TPUG Magazine* has in previous issues covered the general area of accessing local Bulletin Board Systems — an activity that in large urban areas can indeed be quite rewarding (and/or frustrating, depending on the volume of local usage). In contrast to these single-user BBSs, the Delphi information service is accessible on a world-wide basis, capable of handling a large number of simultaneous users. This type of service naturally costs money — the exact amount depends on the system selected, the time of day and the transmission rate.

Of the major 'public' oriented communications services (often called 'hosts'), much has been written on the wonders of the 'big' ones, like CompuServe (located in Dayton, Ohio) and The Source (Virginia). Lately, though, I have come to enjoy the friendliness of several new services — the (Canadian) Bell System's ENVOY, Unison (Denver, Colorado) and Delphi (Boston, Massachusetts).

Those of you lucky enough to live in a city providing direct access to any of these hosts are indeed fortunate, but for the rest (and most) of us, the mere thought of calling half way across the continent to access 'another' BBS would, on the surface, appear ludicrous. However, the fact that these communication services are a *very* long distance phone call away from most users is no longer a problem, thanks to data packet-transmission services, which are now available in most centres of major telephone activity.

The Delphi Connection

The Delphi system has recently asked TPUG to form a Special Interest Group (SIG) on its service. This new channel of communication amongst TPUG members, and between members and the TPUG office, should be up and running by the time you read this. So in this article I want to talk about how you can get on to Delphi, what you can expect to find there, and how to get started with the system.

Delphi may be accessed through local data packet-transmission services from virtually all major centres. Canadians will make use of the Canada-wide DataPac system, while to American users, either Tymnet or Uninet provides the 'gateway' to Delphi. No matter where you live, merely look up the appropriate services in the phone book.

We'll start off with Canadians, because they have the most to do. (Note: if you live in Toronto, you can probably get through on the Delphi direct number, 881-8651. Skip to step 8.)

1) Initiate your phone call to DataPac (from Toronto dial 868-4000 for 300 baud service, and 868-4001 for 1200 baud).

2) When a connection has been established (carrier tone and all that), send a single '.' (dot) if in 300 mode, or '..' (two dots) to signify 1200 baud. When I say 'send', I mean: type a dot followed by **RETURN**.

3) DataPac will answer you with a confirming message: DATAPAC: and a series of numbers.

4) At this point, should you be expecting to converse in full duplex mode (standard with most terminal packages), send the command **SET 2:1**. You won't see this entry if you're already in full duplex, because DataPac thinks you're still in half duplex, and don't really need to see what you've typed.

5) The cursor will drop down several lines and wait there for you to enter an access number. This is the access number of the specific service you wish to reach. In the case of Delphi, this number is 13106 (for purists, this isn't really Delphi's number, but that of Tymnet!). Datapac will now confirm your connection:

DATAPAC: call connected to 1 3106
(xxx) (i, n, remote charging, packet size: 128)

6) The cursor will again drop several lines and you will see the prompt: **TYMNET: PLEASE LOG IN:**

7) If you were to access Delphi through Tymnet in the United States, this is where you would enter the dialogue. Answer: **DELPHI**

8) You will now be asked for your Username, and after that your Password.

At the time of writing (June), we don't yet know the current Initial Signon Username, nor the current Initial Password. With any luck, the magic words should be decided in time to appear in the Inside Information section in this magazine — look there.

Okay so far? Good, let's get back to Delphi. Once past the Username and Password hurdle, wait a moment or two, and voila! the magic of telecommunications is before us — there is the Delphi computer (a VAX, for the curious) greeting us in the name of General Videotex Corporation.

New Users

On entering the required words, you will be (on first access) transferred to the Signon area, at which point you and Delphi become better acquainted, exchanging names, passwords, plastic money and necessary technical details to carry on an intelligible conversation.

New users are usually allowed a certain time (two hours, I think) of access to the system, during which time Delphi is arranging all the necessary registration and posting (yes, as in Post Office!) such confirmations as are necessary.

Signing on to Delphi directly: No matter where you are, if your budget allows, you certainly have the option of calling direct to Boston (actually Cambridge, but what the heck!). The direct-connect (and, I'm told, *much* faster) mode number is (617) 576-0862. When carrier connection is established, enter two **RETURN**s. Then proceed to step 8 above.

Signing on via Tymnet (USA only): Dial your local Tymnet number. If you can't locate one, call Tymnet at 1-800-336-0149. When the request to PLEASE TYPE YOUR TERMINAL IDENTIFIER (or a series of random characters) appears, type **A**. You then proceed to step 7 above.

Signing on using Uninet: (USA only): Information on this system is at the moment unavailable to the writer.

If you *really* encounter difficulty, use the following toll-free number to get directly to Delphi (USA only — Canadian 800 numbers are different!): 1-800-544-4005. If you live in Canada

Communicating on Delphi

by Naomi Epstein

Delphi is a multi-user information system. This system will allow a user to access information on money matters, stocks, education, or flight information; to enjoy challenging games, electronic bulletin boards, an electronic encyclopoedia; and much more. Delphi has members from all over Canada and the United States, plus scattered users from France, Holland and even Japan. Besides allowing access to the services already mentioned, Delphi allows its users to communicate amongst themselves, using 'electronic mail' or a 'conference' mode.

Conferencing

'Conference' is an option that allows users to 'converse' in real time. When in the conference area, one can either form his or her own chatting group or join an already existing one. Groups can preplan meetings at specific times and dates. The group organizer is able to post the information by using the **SCHEDULE** command. Scheduled conferences can be held on any subject, from theology to rock music. Once you enter the conference area, there are several paths you can follow. Except for **SCHEDULE**, all of these options are shown on the Conference Menu.

The first command on the menu is **WHO**. You can use the **WHO** command to view a list of all active groups and their members, all 'idle' conferencees (those not in any group), any users being paged and, finally, a full online list, with those members in conference clearly marked. The members must be marked, because the usernames may not be the same as the names of the conferencees. This refers to the **NAME** command, discussed below.

After typing **WHO** and finding a group that you wish to participate in, the next step is to join it. Just type **JOIN** plus the name of the desired group. Each group has a number as well as a name; if you type **JOIN** and hit **RETURN**, you will be asked for a group number.

If there are no groups that you find interesting, you may **PAGE** another user from outside the conference mode, and form your own group with him or her. Type **PAGE** and the username.

Every user has a 'username'. This is the name with which you sign on. However, while in the conference mode, you can give yourself a nickname, or pseudonym, using the **NAME** command. Type **NAME** plus the desired nickname. This is a temporary change only, lasting until you change it again or leave the conference area.

Immediate commands

Once you are in a group chatting, there are various 'immediate commands' you can use. These are available only while in conference.

If you are paged, for instance, you have two options: to accept or reject the page. To accept it, type the command **/ACCEPT**. This command will automatically take you to the conference area and put you and the user who paged you into a new group. If the person who paged you is already in a group, you will be put into that one. If, however, you do not wish to converse just now with the person who paged you, the command **/REJECT** sends a pleasant refusal on your behalf.

If you wish to page another user while you are in a group, use **/PAGE** instead of **PAGE**. You will be asked whom you wish to page. If the page is not answered and you wish to terminate it, just type **/CANCEL**.

When a user first organizes a group, it will have a number, but no name. In order for all other users to understand what your group is about, it is necessary to assign it a name with **GNAME** followed by the desired name. All present group members will be notified of the group name immediately, and other conference users will see the group name on the online list.

Participating in an online conversation may take a little getting used to, as messages from other users may be coming in while you are typing in your own. You'll adjust to this fairly quickly, however. When you have finished typing in your message, press **RETURN**. Normally, your message will be retyped neatly on your screen (and on the screens of the other participants) with your name in front of it. If you don't want to see your own messages retyped you can turn off the echo with the **/REPEAT** command. The same command can be used to re-enable the echo later on.

While in a group, it is possible to send a private message to any user currently on the system. This is done with the **/SEND** command. This command works like this: '**/SEND** JOWBLOW Hey Joe, come join our conference group. We're discussing rabbits!'. This is a convenient command, because it allows you to talk to any other user, not just those with whom you are conferencing. However, if you are out of conference and do not wish to be disturbed by sends or pages, use the **/GAG** command. When you are in conference, if there is only one user from whom you do not wish to receive input, the **/SQUELCH** command comes in handy. This will turn off input from the user specified.

When you first join Delphi, it will be helpful for other users if you produce a profile of yourself, providing information on such matters as type of computer used and personal interests. The **/WHOIS** command accesses this information. When coupled with a username, this command will reveal all of the information in the desired user's profile. If a user is using a nickname, it is necessary to know their username to view the profile. This is done with the **/RNAME** command, followed by the nickname. The username of the person will be given.

Electronic mail

Another communications mode on Delphi is the 'mail' option. The mail area allows you to send long messages to any user on the system. This user will receive your message when he or she signs on. Since the conference area and the mail area are quite similar, it is possible to go straight to the mail area from conference. Typing **/MAIL** accomplishes this. You may send, read or delete mail from here.

Delphi offers many services besides conferencing and electronic mail. However, these features are likely to be among the most popular and, for many users, will alone justify the relatively small cost of connect time. □

(or Massachusetts), call Delphi direct at (617)-491-3393. Remember, these numbers only apply during normal working hours (Eastern Time).

Now that we're ready to browse the Delphi system, let's take a quick tour from the Main Menu, which reads as follows:

Bulletin Boards	**Merchants' Row**
Conference	**News-Weather-Sport**
DELPHI Oracle	**Office Manager**
Financial Services	**Special Interests**
Games	**Travel**
Groups and Clubs	**Workspace**
Infomania	**Using DELPHI**
Library	**HELP**
Mail	**EXIT**

The next thing you'll see is:

MAIN> What do you want to do?

At this point, merely enter the topic of interest, or at least its first significantly different letters. **B** will get us to the Bulletins Menu, for example, while to get to the Groups and Clubs area, one must enter **GR** to differentiate it from Games.

This being your first time on the system, it would probably be a wise move to see what help is available on Using DELPHI. Type **U** (or **USING**) and you'll get a new menu:

Advice from DELPHI	**What's New on DELPHI**
Credit Policy	**Rates and Prices**
Feedback	**Peoplenet (IamWhois)**
Guided Tour	**Settings (PROFILE)**
Mail to SERVICE	**Telex-Codes**
Manuals Usage	**History**
Network Info	**HELP**
Premium Services	**EXIT**

USING-DELPHI>(Please Select an Item)>

Well, as you see, this can continue for some time and, to the uninitiated, it can be a somewhat bewildering experience. Like a good adventure game, though, as you become familiar with the structure of the system, you find shortcuts and quicker paths to get to those areas that interest you. And you'll perhaps have noticed that there is usually some way to ask for HELP at most points in your journey. (Even if not mentioned in the menu or prompt, entering a **?** will in many cases also get you some measure of help or explanation.) Delphi also provides a complete and very informative manual as an option on joining the service. At something like $19.95, its purchase is highly recommended — the many hints and helps provided will save that much and more in on-line-charges.

Let's exit this menu at this point, for we'd like to introduce you quickly to the Groups and Clubs menu. At the prompt, enter **E** (or **EXIT**, or **CTRL-Z**, which usually will get a result equivalent to exit). Back at the Main Menu, we select **GR** to indicate Groups and Clubs. The Groups menu comes up, and we find there '*FLAGSHIP* Commodore' — obviously of interest to TPUG Members. Now, we all expect that by the time you read this, either in the Groups and Clubs or the Special Interest category you will find a TPUG listing as well. At present, however, *Flagship* is where the Commodore 'action' is to be found. We hope you'll enjoy them both. □

Announcing a New and Practical Application

ComputerEyes™

What can be done with **ComputerEyes**

- Make Custom T-SHIRTS, Scarfs, etc.
- Computer art
- Exchange "digitized" pictures
- Pattern Recognition
- Robotics
- Literally hundreds of practical applications
- Capture real-world images
- Plugs into the Game I/O port
- Save "pictures" to disk
- Print out all images with such packages as; PRINT SHOP, DOODLE, NEWSROOM, KOALA, etc.
- One year parts and labor warranty
- High Quality B/W video camera also available
- Full software support
- Join "free" public domain ART GALLERY exchanges
- **ComputerEyes** interfaces easily to any source of standard NTSC or industrial video. Including:
  - Video Tape Recorders
  - Video Cameras
  - Videodisks
  - TV Receivers with video outputs
  - Other computers

YES! I would like to make computer portraits.

YES! I want my computer to "see".

YES, Please send me **ComputerEyes** as follows:

_ C-64 **ComputerEyes**	**$109.00**
_ Doodle Compatible Software	**11.00**
_ Print Shop Compatible Software	**11.00**
_ Koala Compatible Software	**11.00**
_ Flexi-Draw Compatible Software	**11.00**
_ PANASONIC/IKEGAMI Commercial grade B/W video camera with interface cable	**139.00**

"UNDERWARE" ribbons (to make iron-on transfers)

_ Black (for Epson MX 70/80/82, Okidata, Star Gemini)	**12.00**
_ Color Pens (set of 5)	**12.00**
_ Jumbo Color Pen Set (set of 7)	**15.00**
_ Iron-on Coloring Kit (Contains 20 Sheets of transfer paper and a set of 5 pens)	**15.00**

FOR VISA/MASTER CHARGE ORDERS CALL Toll Free 1-(800)-523-2445, x48
(Pennsylvania residents call 800-346-7511, x48)

OR Send CHECK/Money Order and all inquiries to:
HAL Systems, P.O. Box 293, Scotch Plains, NJ 07076

Please Note: For COD Orders, add $5.00; NJ Residents calculate 6% tax.
Prices include shipping and handling.

YOUR NAME _____

ADDRESS _____

_____

HAL Systems, PO Box 293, Scotch Plains, NJ 07076

*****Dealer Inquiries Invited*****

CompuServe: What Is It?

by Louise Redgers

CompuServe is an electronic networking system that will allow users from all over North America, as well as many other far away places, to talk to each other via their computers. Offering many services to businesses, CompuServe also has a lot to offer to the hobbyist and home computer user. These services range from an electronic mail system that lets me write a letter in Toronto and send it to Seattle, Washington, and have it arrive in the receiver's electronic mailbox twenty minutes later; to taking part in on-line conferences, on every topic from organic farming to science fiction.

Perhaps the most downplayed feature the system offers is the making of new friends. One tends to get on the system and find people of similar interests in the forums. Forums are special interest groups on specific topics. They offer message boards for those who have questions and those who care to provide answers, as well as on-line conferences that allow you to chat with other users and often attend question-and-answer sessions on timely subjects, with respected authorities. It often becomes an on-line party. Hundreds of these forums are currently on-line, many on computer-oriented topics. TPUG has three forums up and running now, and a fourth will be added shortly.

The forums TPUG operates are known as the Commodore Programming Forum, the Commodore Music and Graphics Forum, and the Commodore Education and Entertainment Forum. The titles are broad, but within each of these forums are ten subheadings for both messages and public domain software that can be downloaded, if you desire. The names may be a bit misleading, but let me explain.

The programming forum contains information on virtually every computer that Commodore has made. The B-128, VIC 20 and PET are all supported here. Memory maps, utility programs and many helpful hints can also be found here. At the same time, the C-128 and the Amiga are also developing strong presences on this forum. It is the breeding ground for technical and how-to questions at all levels of programming expertise. The Music and Graphics Forum, while containing hundreds of sample programs, also offers advice on how to program music and graphics on the Commodore 64. The Education and Entertainment Forum offers educational programs for all levels and straight entertainment programs. This forum is heavily used for downloading, but we are trying to encourage further discussion and participation by the users.

An on-line service can be expensive to learn on, if one is not aware of the time spent performing various tasks. However, there are shortcuts for almost everything except Conferencing and Downloading. To reach the forums from the first menu after signing on to CompuServe, type **go pcs ###** to go directly to the desired area. Once you have a very basic knowledge of the system, get rid of the menus; and when you get stuck, type **?**, to get prompts to aid you. This will also save you quite a bit of time. The page numbers for our forums are as follows:

- Commodore Programming pcs 116
- Commodore Music and Graphics pcs 155
- Commodore Education and Entertainment pcs 156

Why use this service?

The service provides answers quickly, if one is stuck with a programming problem. There are so many users from whom to obtain answers, that other users may be able to help on the spot. Large volumes of software to download offers specific task software to those who seek it; if price is a sensitive issue, then public domain might do the job. A link to those who live in out-of-the way places and cannot get to local computer club meetings is perhaps the best feature of the system, as it allows those folks, in essence, to attend meetings, read noted speakers' opinions on various issues, and even ask questions of the experts and manufacturers who can be found on-line.

Why TPUG went on-line

TPUG has almost 16,000 members scattered all over the world. We cannot hope to talk to all of them during the course of a year, nor can we hope to correspond with them, other than through our magazine — the sheer volume makes this impossible. While we were attempting to find more ways to provide more services, this opportunity presented itself, and we took it. This way, our members from Alaska to Newfoundland to Florida can talk to each other and help each other. In some ways it is like having a club meeting for everyone, every night of the week. We can also let other users find out about the club and the magazine, and they will have the information they need, if they wish to join the club. This is just a small part in our campaign to get closer to our more remote users, so that we are more than just a magazine and disk subscription service.

More on-line service

Other than mere computer information, the services of CompuServe range from Science Fiction forums to the ability to make travel arrangements and get weather reports for distant areas on-line. The CB Forum is perhaps the closest thing to dropping in at the local bar on Saturday night. Filled with friendly people who want to chat and have a good time, the CB Forum is truly a party. The Fun and Games area offers you the chance to play with others or against the computer, every game from blackjack to football. The Club area offers a variety of forums for all, from photographers to gardeners.

How to get on-line

In order to get on-line, you will require a Personal Project Number (an ID number) and a password. These can be found in any CompuServe Information Package. The ID number and password identify you on-line, and are usually accompanied by a few free hours of time on the service. This means that you have a manual and some time to experiment before you begin to run up a charge on the system. When you sign on, the system will ask you for credit card information so that it can bill you for the time you spend on it. Time is charged by the minute, at rates of $6.00/hour for 300 baud during off hours, (6:00 pm to 8:00 am EST); and $16.00/hour during prime time. A surcharge applies to 1200 baud users of $4.00/hour. The system currently does not operate at any other baud rates. The Compu-

CompuServe Chooses TPUG

by Louise Redgers

When corporate differences erupted between CompuServe and Commodore Business Machines during recent contract negotiations, CompuServe looked to TPUG to manage the three existing Commodore Forums, add to them and begin a fourth, with our magazine on-line. There has been a lot of excitement around the TPUG office as we prepared to take on this new venture.

Managing a BBS is one thing: running forums on CompuServe is quite another. Without knowing what to expect, and a with a little training, we made our debut on August 30th. We had not been able to preview the forums as sysops beforehand, so we suffered through an educational period, hopefully not at the users' expense. With the help of a good many friends of the club and a few old Commodore sysops, we managed to get through those first few weeks. Now, almost a month later, things are settling down — but not without our having to learn to deal with the unexpected and the unexplained.

Running forums on CompuServe is rather like being the editor of a large daily newspaper. The constant barrage of messages makes it difficult to stay on top of the news. Egos have to be considered, and there are literally hundreds of them. One is constantly aware of the threat of editing a little too heavily, thereby robbing the user of his freedom to express himself; yet one has to be aware that what does not offend one user may easily offend another. The task of juggling these responsibilities has been taken on by a number of courageous souls whom I hope you will get to know better on-line as time goes by.

Our sysop team

Gord Campbell: Gord looks after answering the constant flow of programming questions on the Programming Forum. Gord is the editor of several Canadian computer publications, including *InfoAge Magazine*, *Computing Canada* and *Computer Dealer News*. A long-time user of Commodore equipment, Gord owns a SuperPET, a VIC 20 and a C-64. He also expects to acquire an Amiga in the very near future. As a programmer who also works in the publishing field, he has a way of conveying the tricks of Commodore programming to others.

Gary Farmaner: Gary is one of the authors of the terminal software known as **ProTerm 64**. He is a veteran BBS user with a good knowledge of machine language programming. Currently a university student, Gary has decided to spend his spare time on-line, aiding the users. His current stalking ground on CompuServe includes the Music and Graphics Forum.

Darrell Grainger: Darrell is a constant BBS user, and he has taken his own system down in order to spend time on CompuServe. Darrell is our upload/download expert. As the one with the most spare time and the 1200 baud modem, he kindly volunteered for this tough job.

Tim Grantham: a familiar name to the readers of *TPUG Magazine*. Tim is our music expert. A creative Commodore 64 enthusiast, Tim is helping others to get more out of their SID chips. Tim is an actor, playwright, musician and expectant father.

Betty Knight: Betty has been a sysop on CompuServe since the early days of Commodore's presence on that system. She retired her keyboard about a year ago, but agreed to bring it out of mothballs for TPUG. With her thorough knowledge of CompuServe, her programmer's training and everlasting patience, she has helped us stay afloat when we felt we were drowning in an overwhelming sea. Betty spends time on all of the forums attempting to answer the users' questions on a wide variety of subjects.

Roy Reddy: Roy, a former CompuServe sysop, is a recent addition to the team. Roy, who works at Desktop Computers in Richmond Hill, Ontario, knows a fair bit about all Commodore computers, but his current interest is the Amiga. He is a sports enthusiast who is currently rooting for the Blue Jays to win the World Series.

Carol Shevlin: Carol currently resides in the Education and Entertainment Forum; or rather, where the adventure games are. Carol, who owns a SuperPET and occasionally uses a C-64, is a self-confessed adventuress. Programming in several languages and cleaning cat hairs off her computer are Carol's favourite pastimes — when she is not working as a Certified General Accountant for a local insurance company.

Louise Redgers: lastly, I reside on the system writing those endless bulletins, answering those official questions and trying to socialize with the users, while patiently awaiting the arrival of my Amiga and trying to wear out my C-64, so I have an excuse to purchase yet another computer.

CompuServe chose TPUG because we could offer the talent, enthusiasm and variety of machines and users to the sysop team. We have a ready-made public domain library to share with the users (though sheer volume means that this will only be done a little bit at a time, and much of the material may never make it, due to space limitations and the limited-interest scope of some programs). CompuServe allows us the opportunity to let the club grow even further, share the knowledge that is one of the main reasons for our existence, and hopefully enlarge our library through the donations of software made through the system. It is nice to know that someone 3,000 miles away can leave me mail for next day delivery and reply. What a way for TPUG to reach its members, while showing those who are not members what we can do for them!

See you on-line! □

Serve Information packages are available from your local computer stores: Computerland, Waldenbooks, Sears and Target Stores, to name just a few.

What is Conferencing?

When two or more people wish to talk to each other on the service, they are having a conference. This is the kind of activity we encourage every night of the week. It means that you don't have to leave messages to get answers and to make new friends, but rather you can look and see if anyone is on-line and ready to chat. To check this out, type **users** at any menu, and it will tell you who is on-line, and where they are in the system. 'Access' means that they are uploading or downloading, 'SIG' means that they are reading or leaving messages or bulletins, and 'CO' shows a channel number where people are in conference. Upon entering a forum, the system tells you how many members are in conference. Typing in **CO** at the menu will take you into this electronic conversation station on channel 30. Special events in our forums are held on Channel 1. Read the conference bulletins to learn of upcoming special guest speakers and topics of interest. We are currently attempting to schedule events such as Liz Deal on the B-128, Len Lindsay on COMAL, and a visit by Jim Butterfield. Bulletins will be posted well in advance, to give all users a chance to attend.

What about the TPUG BBS?

The TPUG BBS will continue to operate as usual. It has limited access because of long distance charges and the fact that, currently, there is only one phone line into the system. We encourage members to continue to use it, to find out about local events; and as an inexpensive source of help and contact with other users. □

The Guide

A Monthly Publication For Commodore™ Owners

Commodore support with a twist . . . Personable and even humorous . . . Timely news . . . Helpful tutorials . . . On-going support for several languages: BASIC (including BASIC 7.0 as featured in the 128 PC), Machine Language, COMAL, and Pascal . . . Program Listings . . . *Honest* software reviews.

The Guide features some of the best computer humorists to be found.

- Introduce your "widow" to the Computer Widow's Compendium.
- Tutorials and feature articles by the famous Mindy Skelton.
- Featuring Shelly Roberts' "I'm Sorry . . . But I Don't Speak Hexidecimal." Discover why Shelly just may be the Andy Rooney of the computer world!

We feel we have assembled one of the most talented staff of writers in the Commodore world. Receive each month the most friendly and helpful Commodore publication available. Written by Commodore users who are writing to you, not down at you.

Limited offer — FREE With Each Subscription !

Subscribe *NOW* to capitalize on free software offer!

Receive up to $99.85 in *high quality* software!

With each year's subscription (or renewal) ordered, receive your choice of the software packages listed below, including the award winning educational games from Disney, or Omiterm Terminal written by our own Bob Richardson.

Subscription	Rate	Canadian *	Receive FREE
One Year	$18.00	$24.00 (U.S.)	Any software title listed below
Two Years	$35.00	$47.00 (U.S.)	Any two of the titles below
Three Years	$48.00	$64.00 (U.S.)	All three of the titles below

* Note: Canadian (and other foreign) subscribers must add $2 (U.S.) per program title to cover shipping and handling. No handling charge for U.S. subscribers. Canadian rates listed in U.S. dollars — send U.S. funds only, please.

Donald Duck's Playground

CES Software Showcase Award Winner! — Disney animation at its best! Children play four games to "earn" money to buy playground equipment. Builds money handling skills. Superb graphics. A bestseller! $39.95 retail value.

Winnie The Pooh In The Hundred Acre Wood

— Players explore the Hundred Acre Wood to find lost articles like Ow'ls books, Pooh's honey pot and Eeyore's tail, and return them to their rightful owners. Cheery music from the Disney movie caps off this computer rendition of the beloved classic. $39.95 retail value.

Omiterm Terminal Program

— Written by *The Guide's* own Bob Richardson. Fully supports the new 1660 Modem 300! (The *ONLY* commercial terminal package that currently fully supports the 1660!!!

- **Modem controls accessible from the keyboard**
- **Punter protocol — upload & download — 300/1200 baud**
- **Ten programmable function keys**
- **15 number phone directory**
- **20k receive buffer**
- **Tone or pulse dialing**
- **Auto dial/re-dial**
- **Half/full duplex**

A $19.95 retail value.

The Guide
3808 S.E. Licyntra Ct.
Portland, OR 97222

Don't delay — mail today! Supplies are limited. Offer expires November 30, 1985

Name_____

Address_____

City, State & Zip_____

Enter My Subscription for:
☐ 1 Year ☐ 2 Years ☐ 3 Years

Please Check One: ☐ New ☐ Renewal

You are entitled to one FREE software package, as described above, for each year's subscription purchased. Please list your second choice, as supplies are limited on the Disney titles! Please allow six to eight weeks for delivery.

My first choice is:_____

_____

My second choice is:_____

_____

Sound Advice

by Jim Butterfield

Turn on your brand new Commodore 128 and type the following command:

```
play "u9v1t6hev2t4gv1dv2
  fwv1cv2qeghemu0"
```

The computer will cheerfully play the first two bars of *Three Blind Mice*, in harmony and with a variety of instruments.

This kind of thing has never before been built into a Commodore computer. Sure, they play music, but only with the help of **POKE**s that you need to carefully look up. Although the line above may look like gibberish, it's much closer to writing 'real' music and calls for much less mechanistic work by the programmer.

To translate the above string: *U9* means 'set the volume to level 9'. *V1* says, 'the next bit is for voice 1'. *T6* means 'set instrument 6' — a harpsichord. *H* means that the notes that will follow are 'half notes', moderately slow. *E* means 'play E', the first note of the tune.

. . . I sometimes worry when a computer does so much for you that you lose touch with how it all works . . .

V2 means 'switch to voice 2', and *T4* sets the instrument type to 4, similar to a flute. We play a G, which is the harmony that goes with the first note.

V1 switches control back to voice 1. We play D for the second note, then *V2* switches to voice 2 and plays F. *W* sets the timing to 'whole notes', which are quite slow. At this speed we play the final *V1* (voice 1) note of C. Switching to voice 2, we select the moderately fast quarter note with *Q*, to play E and then G; and go back to a half note with *H* to play the final E. Command *M* says: "Now wait until it's all finished", and *V0* sets the volume down to zero.

Isn't it nice to have a computer do all this for you, rather than the elaborate **POKE**s you had to remember on the Commodore 64?

Synth Sample: Some Sources

by Jim Butterfield, Toronto

Synth Sample, the music program for the Commodore 64, has been something of a mystery. It was placed in the public domain, but the author wouldn't identify himself: many months elapsed before we discovered that George Feil did it. It was written as a demonstrator for a music editor program, but as far as I know, that program is not yet commercially available.

It's a robustly-written program. Once the music starts, you may stop the program; the music keeps playing. You may **LIST** or perform calculations; the music keeps playing. **SAVE** and **LOAD** slows it down a bit, but the music usually doesn't stop.

Here's another good trick. The program loads in one shot; no bootstrap, no data files. That means that BASIC is followed by the music data and by the playing program. When a program is written in this manner, with BASIC followed by other material, it's easy to **LOAD** and **SAVE**. Usually, however, this means that you have two major limitations. First, BASIC must occupy a fixed position: such programs generally won't work if you moved the start of BASIC to make room for a graphics screen, for example. Secondly, you must not change the BASIC program, since such a change would move the following material up or down; few machine language programs or data tables can survive such a move. Surprisingly, **Synth Sample** will allow loading to unusual places, and will survive changes to BASIC quite nicely. That takes planning.

I've gotten letters asking where the sheet music for some of the selections can be found; what OMITD (selection 9) stands for; and (from Europe) what is meant by 'Stationary Ark'. I wasn't able to answer them at the time, but I've subsequently discovered that Mike Martinez has been doing research on the songs and has learned a good deal. As a result, Mike has made it possible to provide the following information.

1. *Stationary Ark*: Theme from PBS nature show of the same name.

2. *Saturdays in Silezia*: Song by Rational Youth.

3. *Spiral*: Main theme from album of the same name, by Vangelis.

4. *Tubular Bells*: Song by Mike Oldfield from album of the same name. Also found on *The Exorcist* sound track.

5. *Magic Shadows*: Theme by Harry Forbes from TV Ontario's movie show of the same name.

6. *Theme from Clockwork Orange*: Prelude for Flatte Trumpets from Henry Purcell's opera *The Libertine*, adapted for the movie by Wendy Carlos.

7. *Oxygene II*: One of six major themes from the album *Oxygene* by Jean Michel Jarre, on the Polydor/Dryfus label.

8. *Canon in D minor*: Fugue by Pachelbel. May be found in many classical collections. Used as the theme for the movie *Ordinary People*.

9. *Enola Gay*: Song by Orchestral Manoeuvres In The Dark.

A final note on sheet music: Mike tells me that the arrangments are adapted from the recordings. No sheet music sources were used. □

Yes and no.

I sometimes worry when a computer does so much for you that you lose touch with how it all works. I don't worry a lot; but I do wonder if new programmers might not understand what the machine is up to.

Here's where a problem may show up: a programmer may want to try a special effect, and can't do it because he or she has lost touch with the working mechanisms. Things like 'glissando' (sliding from one note to another) can't be achieved with **PLAY**, and the user will need to look more deeply; or, you might want to do some sound work in machine language, but without BASIC to help you, you don't know how to make even the simplest beep.

When the 4.0 PET/CBM machines came out, they had useful commands and functions such as **DOPEN**, **SCRATCH**, **RECORD** and **DS$**. It made things so easy for the BASIC prorammer. But when I started teaching machine language to the new 4.0 owners, I discovered that they had no idea how such things as **SCRATCH** really worked. I had to go back and teach fundamentals.

Does it really matter? Maybe not. Personally, I feel much more secure when I'm more directly in touch with the machine and its workings. But the advanced commands save time and effort, and I wouldn't think of forbidding their use.

I often advise users to program in 'generic code'. You may want your programs to be transportable from one machine to another. This is especially true in a school or club environment, where programs should ideally be able to migrate from one model of machine to another — where possible, of course.

But if the advanced features do just what you want, go for them. The new Commodore 128 BASIC is so rich that I wonder if most users will ever learn it all. It seems to me more likely that each user will concentrate on the set of commands that match his or her programming interest areas. Some will use the business features, especially **PRINT USING**. Many will use the new structuring features: **DO**, **WHILE** or **LOOP**, **UNTIL**. Some will go for the error traps, some for the number conversion features. A lot of users will latch on to the great new graphics commands, which will make graphics so much easier than on the Commodore 64.

And some users — more than ever before — by using the new commands **ENVELOPE**, **FILTER**, **SOUND**, **TEMPO**, **VOL** and, of course, **PLAY**, will make beautiful music on their computers. □

After the Online Search

by Don Fox

In my article published in *TPUG Magazine* (March 1985), I suggested that I would follow up with details of how to obtain documents listed in online databases. Before I do so, however, some clarification of concepts is necessary.

database types

In general, there are two kinds of database. The normal terminology for these is 'reference' and 'source' database. A reference database is a coherent collection of data that refers the searcher to another source for full information. For example, a library catalogue gives information about books, periodicals, government publications, et cetera, which the library includes in its collection. Based on this information, one can obtain the document itself to read; there are relatively few applications where only the catalogue record is required. Similarly, the databases that give only bibliographic records in most cases only give the searcher the first step in the information retrieval process; he or she must still obtain the document described.

Source databases, on the other hand, attempt to give full information immediately. For example, DIALOG's ASAP Magazine Index contains the full text of articles taken from a number of popular magazines. Naturally, the storage space required per item is greater, as is the cost of printing or downloading each item, given that this cost depends on connect time, but it does save time and trouble.

There are many other source databases available: indeed, largely because of recent decreases in storage costs, these databases are increasing in number and size more rapidly than the older technology that reference databases represent. An area of especially significant high growth is that of financial databases, such as Dow Jones.

Indirect vs direct

It is apparent that, except for the cost, there is no particular problem in obtaining information from source databases. Unfortunately, much information is available through reference databases only, so it is necessary to learn how to access the documents described by them.

Once again, there are two cases: obtaining the desired document through your local library, and ordering them online.

The first step in the former case is to take your printout to the local library, to see if it is held in the collection. If not, do not give up: the librarian can use your printout to obtain the desired document through interlibrary loan. The printout is needed to prevent transcription errors.

The more traditional route of interlibrary loan has one big advantage — it is cheap, relative to the cost of online ordering. A public librarian who is involved in interlibrary loans in Hamilton, Ontario, recently told me that the typical cost of a non-returnable document (a photocopy of a journal article, for example) ranges between three and five dollars. Normally, there is no charge for a loaned item, unless a charge is made by the library that has made the loan. Many university libraries now charge in the vicinity of ten dollars for loans, since university funding is no longer what it used to be. Most public libraries try to avoid these costs, preferring to borrow from another public library, where possible. When it is not, the library will keep within any limit assigned by the borrower.

Typical prices for online ordering range from five to ten dollars (US) per item, with higher prices for 'rush' service or longer articles (more photocopying). Loans are not usually possible unless the document supplier happens to be a library, and most aren't. These charges are additional to the cost of online time. Many document suppliers also produce online databases, but not all database suppliers provide this service. The **Microcomputer Index**, for example, does not provide it, but the manual suggests that documents listed may be obtained from "many other DIALORDER suppliers". DIALORDER is the trademarked name for the online ordering service provided by Dialog.

If you decide to bite the bullet and order online on DIALOG, you again have two choices. If the item you want has been retrieved online from a reference database, you enter a **KEEP n** command, where **n** is the set or accession number of the set or item required. This is followed by the **ORDER xxxx** command, where **xxxx** is the alpha code of the supplier, as listed by DIALOG. The other alternative, in which you have retrieved the reference somewhere else, requires the use of the **ORDERITEM** string command, which requires you to specify both item and supplier (you can use up to two hundred characters). An example is:

```
ORDERITEM CANDOCS 'Searc
  hing Online Data Bases
  ', TPUG Magazine, Marc
  h 1985, p.22.
```

CANDOCS, by the way, is a Canadian document supplier (hence the name) called Micromedia Limited, in Toronto. For the price of 25 cents per page, 7 dollars minimum per article, they will send you a journal article such as the above. The good news is that these prices are in Canadian funds (US dollar equivalents acceptable). The bad news is that you must have a 100 dollar minimum deposit account with them, or be charged a 15 dollar minimum per order. In addition, you pay a surcharge of 50 per cent for 48-hour delivery. This service is available to any purchasers, not just Canadians, and represents a quick way of obtaining much material, such as Canadian government publications or corporate reports of Canadian businesses, which are slow and difficult to obtain otherwise.

As might be expected, the best bargains from US suppliers tend to be from agencies of the US federal government. For example, the GPO (Government Printing Office) will send you government publications for the list price, plus one dollar per item, plus the cost of airmail postage, if requested, as well as a 5 dollar invoicing charge, if invoicing is asked for by the customer (all prices in US dollars, of course). Rush service costs an extra 4 dollars more per item, which means your request is processed within three working days, and all rush orders are sent airmail. List prices of US government documents, like those of the Canadian variety, tend to be very reasonable.

New Developments

For libraries that are in the business of interlibrary loans in a big way, the world is not standing still. Suppliers such as the National Library of Canada or the Canadian Institute for Scientific and Technical Information are providing electronic mail as a speedy way of requesting documents, using the Envoy 100 system of Telecom

Canada. This service provides special 'scripts' to prompt the user for essential information, depending on the type of material being requested.

One large US system not available in Canada uses microcomputer work stations and special screens to expedite the process even more. Unlike the scripts, which assume one terminal transmitting on line at a time, the screen allows the user to input an entire record and check it visually before sending it. This technique is designed to reduce errors still further. It is very probable that such developments will soon appear in Canada, and also will be made available to the occasional user, as well.

Corrections

I made two statements in the March article that need to be modified. I suggested that a modem providing auto-dial was a good idea for accessing electronic bulletin boards. What I meant, of course, was auto-redial. Not all modems providing the former feature also provide the latter, and it definitely reduces wear and tear on the dialling finger. The other change — I no longer reside at the address given in that article. Please send any further questions and comments to me care of *TPUG Magazine*. □

Telecom Canada

by Ian A. Wright

As part of a recent TPUG Communications meeting, Joe McHugh from Bell Canada gave a very interesting presentation on a number of products available from Telecom Canada. Starting from what a modem is and does, Joe quickly moved into an explanation of iNet 2000, Envoy 100 and Datapac — services offered by the various telephone companies across Canada.

iNET 2000

iNet 2000 is designed to direct information searches in English or French through your terminal, using simple directories. Rather than logging on to a succession of data systems, you can access a series of online databases from iNet 2000 by looking through a National directory, in order to find those databases or information services containing material on your topic. To simplify the search procedure, you can build sub-directories. The Organization directory allows you to select services from the National directory and tailor this to your company's specific requirements. The Personal directory keeps on file your selection of most frequently accessed database services — much like a telephone book.

The cost of iNet 2000 varies, depending on how much you use the service. One example of iNet 2000 use would be to access a travel information database to find out if there are rooms available in Miami, and then select the cheapest flight.

Datapac

Datapac is a nationwide switching network with some 14,000 connections from coast to coast, extending to thirty-five countries world-wide. However, Commodore users cannot access all parts of the Datapac system because of special equipment and protocol requirements. Datapac 3304, for instance, supports the IBM BSC multileaving communications protocol at 2400, 4800 and 9600 bps, and if that makes any sense to you, then 'Punter C1' is baby-talk!

The Datapac we use is an ASCII-based, 300 bps dial-up system, for accessing online databases in other parts of the country without paying long distance charges. For example, The Source database is located in West Virginia, so a regular phone call would mean payment of the long distance charges for your access time. Datapac allows you to call a local number (868-4000) to logon to The Source through the Toronto 'node' of this shared intelligent network. As a user, you pay nothing for this service — the bills are sent from Datapac to the system accessed, so that, in our example, the cost would be included in the bill from The Source. The charges vary, depending on which database is used, the time, the baud rate, the amount of data, or a combination of these factors.

Envoy 100

Both Datapac and iNet 2000 can provide access to a third system called Envoy 100, which is an electronic mail service. E-mail allows you to write, edit, send, receive and store messages across Canada via local telephones. Now you can get or send messages from head office to Hearst, Ontario, twenty-four hours a day in English or French — and the recipient can take them at his or her convenience. If your recipient does not have a terminal, you can use Envoy Post, which lets you send a message of up to four pages to Vancouver, and have the postman deliver it on the same or the next business day.

The charges? Again, they're variable, based on a low monthly fee of five dollars plus the number of kilocharacters (1000 characters, including spaces) that have been used. For example, a 150-word message sent to one user would cost sixty-five cents; and the same message sent to two users — one dollar. Sent via Envoy Post, this same message would cost an additional $1.10 for next-day arrival.

For more information on iNet 2000, Datapac or Envoy 100, call **1-800-267-7400**. □

Wireless Computing

by A. Vic Forde

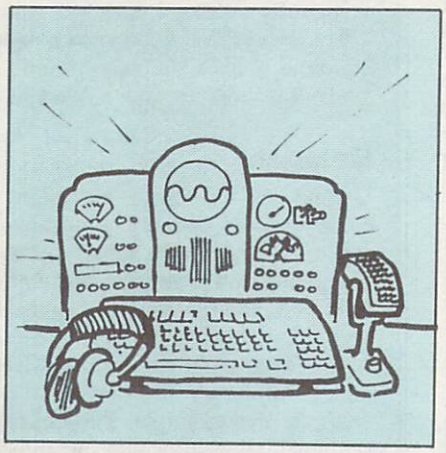
While scanning the January issue of *TPUG Magazine* and reading the list of 'hams', it occurred to me that perhaps many TPUG members might be wondering what that list of names, with its funny-looking combination of letters and numbers after each one, has to do with computers.

The word 'ham' denotes a person who is an amateur radio operator. A ham is skilled in both the science of radio technology and the art of using it for communications. There are two versions of how these amateur radio operators became known as 'hams'. One version attributes the name to an eccentricity of the London accent: "H'amateur radio operators" – later contracted to 'hams'. However, it is more likely that it is an acronym of the names of three amateur radio operators at Harvard University who built and operated a very good little amateur radio station that became well known. They used this acronym – 'HAM' – as a call sign. Pretty soon, the name became associated with all radio stations operated by so-called amateurs.

Nowadays, amateur radio operators must pass examinations set by the Department of Communications, in order to be licensed and to receive a station call sign. The examination requires them to transmit and receive Morse Code at a specified number of words per minute, and to pass a written examination on radio theory. It usually takes someone of average technical skills and education from six months to one year of study before they are ready to take the examination. An advanced licence can be obtained after a period of further study.

The ham radio operator not only has a wide range of frequencies at which he can transmit, but also a variety of modes of transmission available to him. The most common mode used by amateurs is voice transmission. However, many other popular modes, such as Morse Code (CW), Teletype (RTTY), Packet Radio (high speed digital transmission), Television (ATV) and Slow Scan Television (SST), are also used.

In the early days of the development of radio, there were a number of people who began by experimenting with the transmission of very simple radio signals. At this time, the transmission of a radio signal was not controlled by governments, and anyone who had the ability to build a transmitter could freely use one, without restriction. The 'amateurs' who built and operated such a station were very proud of their achievements, and there was much friendly competition among them. The operators liked to advertise their stations on the air to other listeners, so they gave their stations names – the early versions of 'call signs'. Very soon, a substantial number of people around the world were operating radio transmitters.



However, in an attempt to establish efficient use of the airways and known radio frequencies, governments began to regulate and severely restrict the use of this method of communication. One of the consequences of sweeping government control was a regulation stating that each amateur station now had to have a station licence in order to operate. Furthermore, a specific 'call sign' was issued to each station; they could no longer choose and use their own pet name.

Call signs have changed considerably over the years in each country and, at the present time, in Canada, the amateur radio call sign consists of a prefix and a suffix. The prefix indicates the province in which the station is located, and the suffix indicates the particular radio station. For example, Canada has assigned the letters 'VE' to general Canadian stations, with a specific number for each province. The provinces are numbered from east to west. The maritimes are grouped as one province, so the prefix assigned to them is 'VE 1', Quebec is 'VE 2'. Ontario is 'VE 3', and so on, across Canada.

My call sign is VE 3 HPD. The suffix HPD is specifically assigned to my station, and the VE 3 denotes that the station is in Ontario. Ham stations in other parts of the world also have unique prefixes for each country.

Ham radio has been able to survive for the past 60 to 80 years because of the tremendous contributions it has made to radio science, and the immeasurable public service it has rendered to countries in times of national disasters. In recent years, it has reached another threshold in its long history. Hams began experimenting with the marriage of the computer to their radio stations, even *before* the personal computer became available in the market place. They built and perfected interface circuits allowing them to transmit digital signals on the radio frequency spectrum. (This is very similar to the modem used by computers to 'talk' to one another on the telephone lines.) These interface units are now commercially available, and operate in a manner similar to the telephone modem. Instead of the signal travelling along a wire, it is converted in the radio transmitter to a radio frequency, then sent to the antenna.

Amateur radio operators can transmit information to other radio stations anywhere in the world. Furthermore, they don't have an expensive long-distance telephone charge to pay each month.

Amateur radio operators have bulletin board systems similar to the ones familiar to us. They can leave messages for other hams in foreign lands and send messages to and from families of missionaries serving in other countries. In times of national disaster, such as a flood or an earthquake, they can be a vital link in transmitting health and welfare messages to concerned relatives, as well as a major assistance to the Red Cross and public officials. The things that can be done with amateur radio and a personal computer are only limited by one's imagination.

Although the common mode of transmission in ham radio is voice, there are two other popular modes used: Morse

Code, ('CW', in radio terminology); and teletype — ('RTTY'). RTTY is an interesting mode of communication, because it enables amateurs to transmit written text at very fast speeds. In general, telephone transmissions are usually restricted to a maximum of about 300 to 500 baud without the use of special circuits, whereas no such limitations exists when transmitting by radio.

Even though you may not be a ham radio operator, you can still enjoy the thrill of copying teletype if you own or can acquire a reasonably good general-coverage receiver for a few hundred dollars. In addition, you will need to purchase an interface (modem) that will connect your radio to the computer. (This is *not* the same modem you would use to connect a computer to a telephone line.)

With these simple additions to your computer, you will be able to copy on your screen (or to your printer) the major news services located around the world, such as:

1) TASS News Service
2) Associated Press
3) Voice of America
4) Reuters
5) The National Weather Service (USA)
6) The Miami Hurricane Centre
7) METOC — Environment Canada weather
8) ADN — German Democratic Republic News

It is particularly thrilling to copy the news from a news service in the unedited version, containing much greater detail about the various news events. Even more thrilling — you receive this news 'hot off the press', even before it is broadcast over your local TV or radio station.

. . . The things that can be done with amateur radio and a personal computer are only limited by one's imagination . . .

While there are certainly distinct advantages to being both a computer operator and a ham, the computer operator can share in some of the thrills of copying these various news agencies without becoming a ham. If you decide to add this highly enjoyable aspect to your computer, there are a few things of which to be aware. First, the general coverage receiver that you use *must* have good frequency discrimination: that is, it must be able to separate the stations, one from the other. Secondly, you will have to have an outdoor antenna (usually consisting of a long wire in excess of 100 feet). Finally, an interface specially designed for radio communications is required. These interfaces are readily available, and are advertised in amateur radio magazines such as *QST*, *73 Magazine*, or *CQ Magazine*. An interface for receiving only is all that is needed, and these are considerably cheaper than the transmit-and-receive type used by true hams. The price will probably be in the range of 100 to 150 dollars.

With this new-found use for your computer, you can also 'listen in' to ham operators 'talking' to one another in teletype or Morse Code, copying to your screen or printer. You can also copy information and programs from bulletin boards in foreign countries, as a passive 'listener'. Through your electronic 'window', you will truly feel as though you are part of world events as they unfold. □

Next month: How to connect your Commodore 64 to a general-coverage receiver.

Mapping the Protocol Jungle

by Phil Kemp

Users of bulletin boards and other dial-up services allowing uploading and downloading become aware — often painfully — that the terminal program they use must understand a particular file transfer protocol. There are many online services — and many incompatible protocols.

The purpose of these protocols is to ensure that, despite imperfect phone lines, the file is received exactly as sent. To achieve this, we send data characters in groups ('blocks' or 'packets'), with some extra check bytes. We then await a reply, and send the next (or resend the last) block, depending on the reply.

These protocols differ mainly in the length of the transmitted blocks, the way the check bytes are calculated, use of special 'control characters', and how characters are handled that do not have standard ASCII code representations.

The best choice varies according to circumstance. With a high-quality phone line, highest file transfer speed comes with very long blocks, minimizing inter-block pauses. Since errors are few, simple checkbyte calculations suffice. If, however, we have a noisy line, then we expect many errors. More powerful checkbyte calculations are called for. Since a single error requires that an entire block be resent, short blocks are better in noisy conditions to minimize the total amount of data transmitted. Sending data to a friend, we have free choice of protocol; mainframe services offer only limited options.

The most familiar protocol for many of us was devised by Steve Punter, and used on PET and Commodore 64-based bulletin boards. This 'old Punter' protocol has served the Commodore community very well; there are now better alternatives. For most transfers, two printable characters are transmitted, for each byte of the file, so transfers are slow. Precise timing of events is critical. Perhaps for this reason, of the many

File Transfers with KERMIT

by Phil Kemp

Over the last year, there have been many developments affecting Commodore users. In the field of communications, we've seen the price of modems fall; auto-dial/auto-answer modems are no longer rare. Terminal programs, both commercial and public-domain, offer more function and more reliability. For file transfers, I view two changes as most significant: the introduction of 'new Punter' protocol, and a sharp rise in the use of KERMIT.

For file transfers strictly within the Commodore community, the 'new Punter' protocol now provides the fastest and most reliable vehicle. It is used on many BBSs, and is very effective for direct 64-to-64 transfers. However, for file transfers to and from mainframes, and other machines outside the world of Commodore, KERMIT's star is rising. KERMIT is a 'universal' file transfer protocol, developed at Columbia University, New York. It is in the public domain; the university distributes programs and documentation essentially at cost. Many gifted people have contributed programs and ideas; the university continues to act as a clearing-house for developments. There are KERMIT programs in use today on IBM and other mainframes, and on a wide variety of microcomputers. There are at least three programs for the Commodore 64.

Like XMODEM and the Punter protocols, KERMIT is a member of the 'stop-and-wait ARQ block transfer' protocol family ('ARQ' stands for 'automatic request to resend'). File transfer is accomplished by sending blocks of data (referred to as 'packets') back and forth; the sender sends file names, file contents and control information. The receiver acknowledges (positively or negatively) the arrival of each packet, after verifying the content, using check characters contained in the packet.

KERMIT was designed to be simple, reliable, and usable with diverse hardware. Efficiency was a lesser consideration; even so, communicating at 300 bps, KERMIT runs at 50 to 80 per cent efficiency. That is, data is transferred at a respectable rate of some 15 to 24 characters per second.

Documentation from Columbia has evolved through many editions, and is comprehensive and readable. Also, good technical descriptions have appeared in *BYTE magazine* (June/July 84), and the *PC TECH Journal* (January 85).

KERMIT continues to evolve. As well as the original simple one-byte block checksum, there are now stronger two- and three-byte check options. Each transfer begins with exchange of 'initialization' packets. If both sending and receiving programs agree at this point, then variations of the basic protocol may be used. This allows new programs to be used, to exploit extensions to the original base protocol, while ensuring that older programs will continue to work.

In future, we should expect wide use of faster (1200 and 2400 bps) modems. Many of today's popular protocols will not fully exploit the higher speeds, since the time to send a block becomes short while the pauses between blocks remain the same length. In the case of KERMIT, there are many knowledgeable users, and the protocol provides for extensions to the rules, so I expect evolution to take advantage of hardware changes.

KERMIT provides a valuable option for communication with the non-Commodore world. It's one of the very few file transfer protocols that has any hope of gaining widespread acceptance across multiple computer types. Philosophically, there are parallels between KERMIT development, and the cooperative activity of clubs like TPUG. □

public domain programs supporting this protocol, I've yet to see one purely in BASIC. Some commercial terminal programs (**VIP-Term**, for example) support the protocol, but there are persistent reports of total or partial failures of transfers.

Steve's 'new Punter' protocol is used on a growing number of Commodore BBSs, and it has significant advantages. Block length is variable, to a maximum of 255 bytes; short blocks can be used to cope with poor phone lines. Blocks contain 4 check bytes (to a unique formula), a block length indicator and a block sequence number, followed by data bytes. This now provides the fastest 64-to-64 transfer speeds of any protocol in common use — and is likely the most dependable. Good public domain programs are available, able to transfer both to and from a BBS and other C-64s.

Outside the Commodore world, the best-known protocol is XMODEM. It's used on many bulletin boards using the CP/M operating system, and on some mainframe services. Transmitted blocks contain 128 data characters, preceded by a 'Start of Header' byte (ASCII code 1) and a block sequence number, and usually trailed by a single block check character. As with 'old Punter', this is a simple checksum of the data byte values. Some commercial terminal programs support XMODEM. The protocol is 'forgiving' in timing of events; there are public domain programs entirely in BASIC. There is no provision for handling non-printable characters.

From Columbia University, New York, comes KERMIT (see opposite). Designed primarily for use with diverse computer types, KERMIT is (for practical purposes) public domain, and increasingly widely used.

Many other protocols are used in special cases. For example, CompuServe's 'proprietary B protocol'; just buy their program (VIDTEX), and all will (magically) work (unless your access is via Datapac, anyway). Many vendors supply IBM micro-to-mainframe protocols; usually there are no programs available for non-IBM machines, and no documentation to write one.

The computing industry is fast-changing. File transfer is an area particularly short of accepted standards. To make the most of this mess, we need an awareness of the variety of protocols in use, and the major distinctions. Then we can select an appropriate protocol (and, therefore, terminal program) for our needs. □

The C1 Protocol

If you have ever uploaded or downloaded programs on a Commodore BBS, the chances are good that you have made use of one of Steve Punter's file transfer protocols, familiarly called Old and New Punter Protocol. In this article, the first of two parts, Steve explains in detail the theory and construction of his 'New' protocol, officially termed 'C1'.

Inception

During the summer of 1981, when I first got the idea of putting up a BBS, I started work on a simple protocol for transferring programs to and from the BBS. This protocol was similar in structure to XMODEM, and had about the same reliability. Under good line conditions, it would give error-free transfers (this was to be expected). Under moderate noise conditions, the protocol would hold up, and would still give error-free transmissions. It was under poor line conditions that it (as well as XMODEM) would fall apart.

In the summer of 1984, I started work on a very ambitious project: to produce a protocol that was both fast, and *extremely* reliable, even under the worst of line conditions. From this work came the 'C1' protocol; not a simple block/checksum affair, but a complete communication system for the computer.

Be warned, therefore, that understanding the ins and outs of 'C1' will not be easy, but with enough patience, there's no reason why even the least-skilled programmer cannot be comfortable with it.

Concepts

The concept behind the 'C1' protocol was simple: to allow two computers to 'talk' with one another (while transferring data) in such a way that nothing short of a complete distortion of the transmission line could result in a misunderstanding. If this concept could be realized, then files could be transferred between computers without fear of line noise causing a breakdown in the protocol; or of the received data differing in any way from that which was sent.

Nothing is perfect, though, and I don't for a minute claim that 'C1' is completely infallible, but I can say, with reasonable comfort, that 'C1' can deliver bad-line accuracy not found in any other microcomputer transfer protocols. For this accuracy, though, there is a price to pay, and it is *complexity*: the protocol is extremely difficult to duplicate without a complete understanding of its workings.

A Simple Conversation

In first deciding how the protocol would function, I thought of how two people could carry on a conversation under high noise conditions, where misunderstanding would be the norm. The scenario I'm going to give differs from the protocol in that the people talking have no way of verifying the accuracy of what they *believe* they have heard. It is meant to demonstrate how the two computers 'talk' with one another, discussing the necessary repetition, or non-repetition, of each block of data (the cornerstone of a checksum-based transfer protocol).

Ken and John are attempting to assemble a machine in the middle of a very noisy machine shop. Ken reads the instructions to John, who carries them out. Even at close proximity, the two have difficulty hearing one another, so they adopt a form of banter that allows each instruction to be verified and acknowledged. Here is how the conversation might go:

John: "Put part 'A' in hole 'D'."

Ken: "Understood, putting part 'A' in hole 'D'."

John: "Acknowledged, let me know when you are ready for the next instruction."

Ken: "Go ahead, what do I do next?"

John: "Put screw 'E' through slot 'T'."

Ken: "I didn't understand that, could you please repeat."

John: "Oh, okay. Tell me when you're ready for that instruction again."

Ken: "Ready now."

The conversation continues in this fashion, guaranteeing that both John and Ken are fully aware of what the other is doing. In real life, people wouldn't have the patience to keep up that sort of banter, but that's why they make more mistakes than a computer.

It is just this sort of 'conversation' that the two computers have between each other, only the language is different; the instruction is replaced by the block of data, and all other statements by special codes.

Communication Codes

One of the areas where simple protocols fall apart is in the transmission of 'handshaking codes'. It's called 'handshaking' because it implies that the two computers are having a dialogue, rather than a monologue. These other protocols rely on single byte (8 bit) words for their communication codes. That could spell trouble, since the likelihood of any one 8 bit code being transposed into another is greater than for multiple byte codes. For this reason, 'C1' uses 3 byte (24 bit) codes — sufficiently different that the likelihood of a transposition is extremely low. Not only that but, as you will soon learn, the method of receiving 3 byte codes is so designed that if there is sufficient line noise to make the neccessary transpositions, there would most likely be extra characters sent; 'C1' can avoid this situation.

Five distinct codes are used in the protocol: **GOO**, **BAD**, **ACK**, **S/B** and **SYN**. Each has its own meaning, just like any English word, and all are used in a specific sequence so that synchronization difficulties will be automatically identified and corrected.

Checksums

When a block of data is sent, we must have a way of determining if it is correctly received or not. This is accomplished by using what is known as a checksum. Quite simply, a checksum is a number mathematically derived from all the bytes within the block. The receiving computer recalculates the sum and compares it with the sum received along with the block. Theoretically, any fault in the transmitted data will result in the two checksums not matching; but that's theory. In reality, the accuracy of the checksum depends on the type of mathematical operation used to calculate it, and what kind of noise it encounters.

The simplest way to create a checksum is to add up all the ASCII values of the bytes contained in the block. This is fine for many types of errors, but not the type that inverts a particular bit. Should two identical inversions occur on two opposite bits, the sum will remain the same. For

example, let's add the following two bytes:

```
   1101 0011 = 211
 + 0110 1101 = 109
   ---------------
               320
```

Now assume that the fourth bit from the right of both of these bytes becomes inverted by line noise:

```
   1101 1011 = 219
 + 0110 0101 = 101
   ---------------
               320
```

As you can see, the sum remains 320, even though line noise has made obvious changes to the bytes. A better system is one called 'cyclic redundancy', which works on a somewhat different principle. The checksum is 16 bits long, and is created in the following fashion: each byte from the block is Exclusive ORed with the low order part of the checksum. The checksum is then *rotated* one bit to the left, and the procedure repeated with the next byte.

Even this highly superior method can be tripped up, so I have combined *both* an additive checksum and a cyclic redundancy checksum to create one very-hard-to-beat 32 bit 'super' checksum.

Listening for code words

Although 3 byte code words are more reliable than 1 byte code words, nothing is perfect. It has been said that if you let an infinite number of monkeys bash away at typewriters for an infinite amount of time, one of them would eventually type 'To be or not to be, that is the question'. Although this stretches statistical probability to its limit, this kind of thing can easily happen on a smaller scale: the letters **GOO** could quite conceivably be produced by purely random line noise.

To try and eliminate *all* possible errors isn't feasible, but 'C1' makes an attempt at eliminating as many as possible. It is very probable that any noise capable of randomly producing **GOO** would not stop there; more likely, it would produce a string of characters, something like **HGOOEK**. Were we to allow the protocol to listen exclusively for three letter combinations, it would most assuredly pick out the **GOO** in that string.

My specifications for 'C1' call for a code recognition routine that will *only* make code word comparisons on the *last three received* bytes. This is accomplished in my coding by going back and testing for further characters after I have identified a three byte code word. Should another byte be present, the identified code word is thrown away, and the search will continue.

Statement and Listen Loops

One immediate drawback to the system described above is that a *real* code word, masked within some random noise, would be rejected by the receiving computer. This would also be true of a code word simply damaged by noise (like **GOE**). For a protocol to be impervious to this sort of corruption, it must be capable of restating code words over and over until the receiving computer can understand, yet it must also have a way of knowing whether the receiving computer got the code word or not. This was a fact that eluded me when I wrote the original protocol.

When we talk to other people, the cornerstone of understanding is recognition. If we ask: "What do you think?", yet get no reply, we ask again. Only when we receive a reply from the person to whom we are talking do we continue with our next statement. It would be pointless wasting our breath on someone who isn't listening.

Within 'C1', communication between computers is handled through a similar system, which I call the 'Statement and Listen Loop'. It's quite simple, really: when one computer has to 'say' something to the other, it does so, then waits for a predetermined time for a known response. Should it fail to receive a response within that period of time, the code word is said again, and the computer listens for the reply. This continues until the required response is heard.

The system is further enhanced by the fact that both computers are *always* engaged in a 'Statement and Listen Loop'.

Synchronization Lock

That rather ominous-sounding title is actually rather simple: it refers to a condition whereby the 'Statement and Listen Loops' of each computer become locked together. This is analogous to two people speaking at the same time, over and over, so that no effective communication takes place. In order to guarantee that the two computers never get into this state, the wait times of the loops are altered slightly.

Assume that the fixed wait loop time was 0.5 seconds; this is called a 'Short' wait. We also have a 'Long' wait, which would be slightly longer, say 0.6 seconds (actually, the delay within a 'Statement and Listen Loop' is not particularly critical, but should be somewhere in the neighbourhood of one half second). Each time the computer goes through an SLL, a counter would determine which type of wait to use, long or short. The sequence is broken into three: the transmitting computer will use a Long-Long-Short, while the receiving computer will use a Short-Short-Long.

Block Structure

Each block of data contains somewhat more than just a collection of characters taken from disk: it also contains a 'header'. The header is 7 bytes long, and contains the following information:

Byte 1: Low part of additive checksum

Byte 2: High part of additive checksum

Byte 3: Low part of CLC checksum

Byte 4: High part of CLC checksum

Byte 5: Size of next block

Byte 6: Low part of block number

Byte 7: High part of block number

As you remember from the section on 'checksums', there are two distinctly different, 16 bit (2 byte) checksums. One is an additive checksum, composed of the mathematical sum of the PETASCII values of all the data bytes (and bytes 5 through 7 of the header). The other checksum is calculated using Cyclic (CLC) Redundancy (on the same bytes). These 32 checksum bits are placed in the first 4 bytes of the header.

The 5th byte is the length of the *next* block. This may seem odd to some, but consider the difficulties in sending the size of the current block in that self-same block. You need to know the block size in order to calculate the checksum, but you can't know for sure that the block size is correct unless you have verified the checksum — a 'Catch-22' situation. By sending the size of any given block in the *previous* block, the size is known for a fact *before* the checksum is calculated.

The 6th and 7th byte contain the block number. This was added quite early in the development of 'C1', under the assumption that it would be necessary (as it is in XMODEM). As it turned out, 'C1' uses a method of handshaking that makes this unnecessary. Nonetheless, my specifications call for its inclusion, as certain uses of the block number could be made. Also, the high order part of the block number (byte 7 of the header) is used to flag the last block.

Concluded in next issue. □

Micro Processes

A 64K Expansion For The Olde Pette

by John R. Kurczak

Those who are looking for ways to extend the power of their PET 2001 computers may want to consider the 64K expansion RAM board from Microport Microcomputer Services. Along with the 80 column keyboard conversion I discussed in the May issue, this board effectively transforms the 2001 into an 8096. The board is approximately 9.5 inches long by 4 inches wide, much smaller than other boards available for the PET. It mounts very snugly just inside the rear casing, leaving the main logic board accessible to the user. This tiny size is achieved by using eight 4164K RAMs. There are also four empty ROM sockets that are not used. No information is given for their purpose in the board design. If these sockets could be eliminated, the dimensions of the board would be decreased even further. The 6502 processor chip fits into the other empty 40 pin socket.

The installation of this board is more involved than the 80 column keyboard, as there is some soldering and desoldering of the main board. Depending on your ability to solder and the problems encountered, the board should take no more than an hour or two to install.

The extra 64K of RAM in an 8096 is addressed in the second half of memory space in four 16K selectable blocks. The BASIC ROMs must be disabled before the processor can access the RAMs in this area. A **NO ROM** line is required to accomplish this — processor socket pin 5 is connected to all pin 21s of the ROM sockets. Because this **NO ROM** line does not exist on the 2001 nine inch screen PETs, a small modification to the logic board is required.

The installation requires the following steps:

1) Unplug the wires from the main logic board and remove it from the machine casing by pinching the nylon pillars and pulling upwards on the board.

2) On the underside of the board, solder a 120 ohm resistor with attached wire to pin 5 of the 6502 processor chip UC4. Solder the other end to pin 20 of UC3.

3) Cut the wire to a length of 80 mm and save it for step 5. Now solder this wire to pin 5 of UA3 near cassette port 1.

4) Pin 18 at UD4 is desoldered (a desoldering gun is recommended for this by the manufacturer, but I used some desoldering mesh with no problems — much cheaper for a one-time job). The pin must be pulled out from the logic board when the solder has been removed. Clean the hole and bend the pin out from the chip.

5) Strip one end of the wire from step 3, pass it through the desoldered hole and solder it to pin 18 of the chip. Next, solder the other end of this wire to pin 6 of the UA3 chip after it has been cut to fit.

6) Remove the 6502 processor from its socket and place it into the extra 40 pin socket on the 64K board. Make sure the little notch (this tells you which end of the chip is the front) is facing in the same direction as the rest of the chips already mounted on the board.

7) Now the board must be positioned into place on the rear casing inside the PET. The board is placed with the ribbon cable to the right (this plugs into the now-empty processor socket in the main board) so that the cut-outs for the port are not blocked). Press it into position.

8) Refit the main board and then plug the ribbon cable into the processor socket. Fit the remaining cabled sockets and check all the connections.

To test the board, turn on the power. You should get the normal '31743 bytes free' message. If no 8096 software is available, use **POKE 65520,128**, which will crash the machine if the board was connected properly. My machine did crash, and I am happy to say it worked properly the first time.

I did have some problems desoldering the pin in step 4. I am no expert in soldering, and was afraid to apply too much heat to the wire mesh that sucks up the solder. Finally, after four tries, I got the pin desoldered without ruining the chip on the board. This was the only difficult part of the whole assembly. Make sure you have the proper locations of the chips mentioned in the instructions — *no diagram is provided.* Having a diagram would have made things a little easier to locate on the board.

To test its compatibility with an 8096, I loaded the **COMAL 1.02** language and found that it booted up without a hitch. The extra 64K memory gives you 38692 bytes free to program with in COMAL. I have had no problems in a couple of months of use. Now all I need are some word processors and spreadsheets to further test its compatibility.

The 64K RAM expansion board can be obtained for 250 pounds sterling from: Microport Microcomputer Services, 7 Clydesdale Close, Borehamwood, Herts., England, WD6 2SD, Tel: 01-953-8385. □

Write Protect In Software

by Efraim Halfon

Assume that an important program or a data set, stored on a 1541 floppy disk, must be circulated to other users, who willingly or unwillingly may modify the disk content. A write-protect sticker on the disk may be a temporary protection but... the

sticker can easily be removed. The solution is to put the protection on the disk itself. A short program is sufficient to install write protection that is very difficult to bypass or remove. The 1541 operating system (DOS) itself will prevent anyone from writing on the disk and the sticker is now redundant, subject to certain limitations described later on. So, how is this done?

Very simply: all that's required is to change one critical byte on track 18, sector 0 of the disk. This byte (byte 2) normally holds a value of 65 (ascii 'a'), which is placed there when the disk is first formatted. On subsequent write operations (*except* formatting) this byte is checked by the DOS; if it does not contain 65, the DOS assumes that the disk is in a foreign format and will not allow the write to proceed. However, read operations are not affected.

The program below changes byte 2 on track 18, sector 0 from 65 to 63 (ascii '?'). This is accomplished with the **U1** (block-read) and **U2** (block-write) direct access disk commands. The entire block is read into one of the disk drive buffers, the byte is changed, and the block is written out again. And that's all it there is to it. You now have a write-protected disk, and it's all done in software.

Note of caution

Remember, protecting your disks by this method is like getting a tattoo — there's no allowance for second thoughts. While your data and programs can now be widely circulated without you worrying about somebody carelessly destroying them, the only simple way to defeat the write protection is to reformat the disk — which destroys all your data. □

```
10 rem open error channel 15, write c
   hannel 5
20 open 15,8,15: open 5,8,5,"#"
30 rem read the entire block of data
   <space>from the disk
40 print#15,"u1:5,0,18,0"
50 rem set buffer pointer to byte 2
60 print#15,"b-p:"5,2
70 rem change byte 2 to question mark
80 print#5,"?";
90 rem write to disk in track 18, sec
   tor 0
100 print#15,"u2:5,0,18,0"
110 reset drive
120 print#15,"u;"
130 close all channels
140 close 5: close 15
150 rem that's all folks!
```

Changing the 1541 Device Number

by Robert E. Ross

In the Answer Desk section of *TPUG Magazine*'s April issue, Ronald Lisoski brought up the problem of making a 1541 disk drive 'aware' that its device number has been changed by a user-rigged selection switch. His answer was to turn the drive off, then on again. There is a software alternative available in the disk reset command, however. Use either:

```
open 15,<dn>,15, "uj": close 15
```

or:

```
open 15,dn,15, "u:": close 15
```

where **<dn>** is the current device number of the drive. This command works by executing the power-up vector. If the disk command channel is already open, use **PRINT#15,"UJ"** instead. As a point of interest, you can also use **U;** or **UZ**, instead of **UJ** or **U:**, which are the standard forms.

While it is convenient to have a software alternative for resetting the drive, turning the drive off and then back on is probably easier. Those who like to type can use the 'memory-write' command:

```
open 15,<dn>,15
print#15,"m-w"+chr$(119)+chr$(0)+
  chr$(2)+chr$(32+<new dn>)+
  chr$(64+<new dn>)
close 15
```

Memory location 119 in the 1541 contains the listen address of the drive (32+**<dn>**), and 120 holds the talk address of the drive (64+**<dn>**). The two can be different, but BASIC programmers won't find this information very useful. BASIC does seem able to handle drive numbers in the range of 4 to 30, when 119 and 120 are both set with the same value. The **UJ**, **U:** and **M-W** commands are explained on pages 38-40 of the *1541 User's Manual*, 2nd edition. But note that the colon should probably be omitted (**M-W**, not **M-W:**); my drive never does the command correctly when the colon is used. □

20-dollar Software? Here's Why Not

by J. Allan Farquharson

Recently I read in a weekly computer newspaper a letter to the editor that claimed one should be able to buy commercial software for about 20 dollars. Current prices in hundreds of dollars were strictly rip-offs. Perhaps you agree.

Writing a program takes time. Did you ever type in one of those long ones from a magazine? That takes long enough. But programming a big commercial product will take from three hundred to one thousand or more man-hours. Most of these are written by teams to cut down the overall time in getting finished. Apple's Lisa computer cost 50 million dollars to develop, much of it spent on the software products and the operating system.

"I want your program for twenty dollars." A car for two thousand is a good price. Why not? Often the dealer gets up to forty per cent of the sale. Then the distributors take about another forty. That leaves about twenty per cent for you, the guy who wrote the program. If the sale price is one hundred, you get twenty. If the sale price is twenty, you get four dollars. We can argue about the split on price, but you will almost always get the small part!

If you pay yourself thirty bucks an hour, then you need to get back 30 times 300 for that super program that took you 300 hours to write. That's only nine thousand bills. At four bucks a shot that will take sales of 9000/4. Only 2250 sales to get your money back!

Now, let's say that your friend Albert buys a copy and sells it to his friend for fifty cents, and she gives a copy to her friend for nothing. And he gives a copy to his friend... Now your sales dwindle to five hundred. You never get any more because the whole thing is a give-away.

There is another problem, aside from that pirate, Albert, who puts you out of business. You cannot sell anything without it costing something. Let's drive to the big city and spend a month there visiting all the folks you can find. Hotel: 900 dollars; meals: 600 dollars; gas and wear and tear on the old heap: (not you!) 600 dollars. Not bad — and you made forty sales.

Just as a matter of interest on that sale of goods: If you take all the money, you just netted 40 times 20, or 800 dollars. But you spent 2100 dollars, and you haven't paid the rent at home yet, and you're running low on groceries. Something sadly wrong there. I'll bet you a canister of skink oil you know the answer. You *have* to sell the product for more money!

Right! And many other costs will have to be worked into the picture: Advertising, printing, wages, office overhead, telephone, heat, light and power.

Perhaps I'd better let the distributor sell my product. But he has costs, too, and that 20 dollar price is a rocket that will never leave the Cape.

Being a financial wizard is not my thing. Having spent several years developing software, I have discovered some of the realities. And by the way, about that guy in the US who was selling Pascal for twenty-nine dollars for a couple of years — he went bankrupt owing many folks a copy for which they had already paid. And he advertised nationally! I bet his arithmetic was as bad as mine. And that's why 20 dollar deals are not in orbit. □

DATA Dodger

by Paul Blair

The idea for this little routine came from a friend of mine whose work entails a lot of statistical processing. His difficulty was that the arrays he had set up for efficient operation were not big enough to handle all the data in his **DATA** statements at one time. Could he enter all the data, but operate selectively on only parts of it?

BASIC uses a pointer to keep track of **DATA** values. You may have a table of 200 items, and **READ** them ten at a time at various parts of your program. When you **READ** for the first time, the pointer is aimed at the very first **DATA** item. With each **READ**, the pointer moves on to the next value. You may jump back to the start of your **DATA** with **RESTORE**, but there is no way (apart from **READ**ing all the intermediate values) of jumping forward. Or so I thought.

The Commodore 64 and VIC 20 store the **DATA** pointer in locations 65 and 66 (62 and 63 in PETs). By poking the values found at locations 61 and 62 (58 and 59) — the pointer to the address of the current BASIC statement — into the DATA pointer, you can easily set up subroutines to read in any desired block of data.

```
10 REM:DATA DODGER
20 :
30 REM:PAUL BLAIR
40 :
50 REM: TO PERMIT SELECTION OF
60 REM: DATA FROM A LONG LIST.
70 :
500 INPUT"WHICH BLOCK";A
510 ONAGOSUB 600,700,800,900:END
520 :
600 PL=PEEK(61):PH=PEEK(62)
601 POKE65,PL:POKE66,PH
620 READA$,B$,C$:PRINTA$,B$,C$
640 DATA FIRST,FRED,3
650 RETURN
660 :
700 PL=PEEK(61):PH=PEEK(62)
701 POKE65,PL:POKE66,PH
710 DATA SECOND
720 READA$:PRINTA$
750 RETURN
760 :
800 PL=PEEK(61):PH=PEEK(62)
801 POKE65,PL:POKE66,PH
810 DATA THIRD
820 READA$:PRINTA$
850 RETURN
860 :
900 PL=PEEK(61):PH=PEEK(62)
901 POKE65,PL:POKE66,PH
910 DATA"FOURTH"
920 READA$:PRINTA$
950 RETURN
```

Line Delete Subroutine

by M. Garamszeghy

Often when developing a new BASIC program or extensively modifying an old one, I am faced with the task of deleting relatively large blocks of the program (10 to 20 lines or more). Unfortunately, Commodore BASICs prior to the Plus/4 do not include the **DELETE** command found in some more advanced forms of BASIC. The lines can, of course, be deleted by entering each line number followed by **RETURN**, but this becomes very tedious after the first dozen or so lines!

My solution to this annoyance is a short subroutine called **Linedelete**. This routine can be included at the beginning of any BASIC program under development. Once the program is finalized, you can delete the routine. (No, you cannot use **Linedelete** to delete itself.) The routine will not affect the operation of a normal BASIC program if you choose not to delete it.

Lines 0 and 4 form a bypass around the routine when the host program is run.

To use **Linedelete**, simply type the following in the immediate mode, using the appropriate numerical values for the parameters in square brackets:

```
LO=[1st line# to del]:HI=[last line#
  to del]:IN=[line# increment]:GOTO1
```

The program will print out the value of *LO*, followed by a statement similar to that which you just typed in, except that the value of *LO* will be incremented each time by *IN*. This process will be repeated until the value of *LO* is greater than *HI*.

Linedelete works by 'fooling' the computer into thinking that a series of commands has been entered in the immediate mode. (You can only delete lines when in the immediate mode.) This is done by printing the line numbers to the screen, then poking into the keyboard buffer the same characters (cursor ups and **RETURN**s) you would use to delete a line in direct mode. Each time program execution halts, the keyboard buffer is read and the line is deleted just as though you had done it yourself.

While not as fast or romantic as an intrinsic **DELETE** command, this subroutine is equally effective at removing unwanted lines. In some cases, it can also be more versatile than a normal **DELETE** command. For example, if you wanted to delete every second line in a range of the program, setting the *IN* paramater of the **Linedelete** routine to the correct value would bypass the lines you wanted to retain.

The version shown is for the Commodore 64. However, the routine can be easily adapated for other Commodore computers. For the VIC 20, simply insert an extra cursor up in the string *U$* in line 1. For PET/CBM machines, change the assignments of *K* and *N* in line 1 to 622 and 158 respectively. □

```
0 goto 4
1 K=630:n=198:u$="<4 up>"+chr$(13)+
  chr$(13):u=len(u$):iflo>hithen end
2 printlo:fori=1tou:pokeK+i,asc(mid$(
  u$,i)):next:poken,u
3 print"lo="lo+in":hi="hi":in="in":go
  to1":end
4 :
```

How to Beat *Sargon III*

by Don Fox

Despite Mike Martin's warning in the March issue, I recently set myself the task of solving **Sargon III** at level 1 (the default level). By this I mean that I wanted to discover a series of moves that always wins. I have noticed before that chess programs rarely deviate much from a set of moves, if you play the same set against them, and this proved to be the case with **Sargon III**, with certain exceptions I will mention later.

Let me hasten to say at the outset that I have not yet been totally successful in achieving my goal. I have made significant progress, however, and I pass this on to you to encourage any would-be conquerors of **Sargon III** to persevere. While it is a remarkably powerful program, it is not unbeatable.

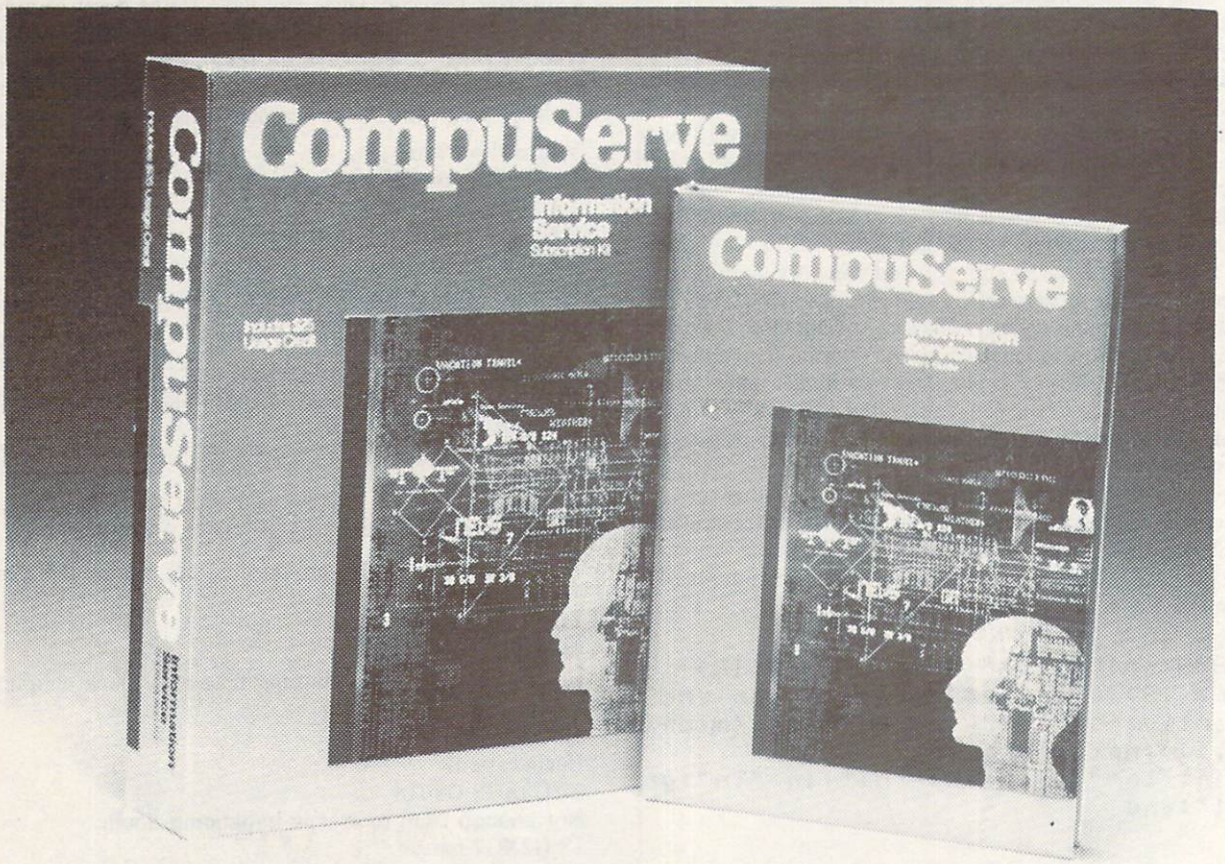
When you boot up **Sargon III**, it automatically gives you level 1, which means it takes an average of five seconds per move. In addition to this, however, the program is busy calculating its next move whenever you are contemplating yours. If you wish to prevent this, you can enter **CTRL-E**, which puts the program in 'easy mode' — the default level is not the easiest one available. This discussion leads to one reason why **Sargon** sometimes deviates from a sequence of moves: if it has been given time to look further ahead it may hit upon a better move leading to a superior position. Once you have found a set of moves that wins, however, as long as you play these at the same speed as you did when you discovered them, **Sargon** should make the same responses, enabling you to impress your friends with your chess playing skill. As an example consider the following game:

White: Don Fox
Black: **Sargon III**
1 D2-D4 G8-F6
2 E2-E3 G7-G6
3 F2-f4 F8-G7
4 B1-C3 O-O
5 G1-F3 D7-D5
6 F1-D3 C8-F5?
7 D3XF5 G6XF5
This leaves Black's King's position seriously weakened.
8 O-O B8-C6
9 F3-E5 F6-E4
10 C3XE4 F5XE4
11 D1-G4 C6XE5
12 F4XE5 D8-C8
Black wishes to exchange Queens, but White declines.
13 G4-G3 C7-C5
14 C1-D2 C5XD4
15 E3XD4 C8XC2
Here White has set a cunning trap and Black obligingly falls in.
16 D2-H6 C2XG2+
Desperation!
17 G3XG2 G8-H8
Not enough time to escape impending doom.
18 G2XG7 mate

This game is forced after Black's first move. You can compel Black to make this move by entering **CTRL-V** at the beginning of the game, making the first move for both White and Black, and typing **CTRL-V** again to free **Sargon** to play Black's moves. If this seems like cheating, you can let **Sargon** move freely from the outset and the program will frequently make the desired move anyway, since opening theory regards it as a good answer to **1 D2-D4**. The complete solution would force mate no matter what Black's first move was but, as I mentioned above, I have not yet achieved that. I have also discovered a 15-move forced mate at level 1, starting with **1 D2-D4; D7-D5**, another opening approved by standard theory. This win appears to result from a bug in **Sargon III**, and I will not reveal it here, but rather leave it as a challenge for the reader. If you're not up to that, you may want to try some variations on the above game; there is one that leads to mate in 16, an improvement of 2 moves.

I would also welcome any games from readers that despatch **Sargon III** in fewer than 20 moves starting from other common openings, such as **1 D2-D4; D7-D6**, **1 D2-D4; C7-C5** and **1 D2-D4; F7-F5**. Note that I have specialized in Queen pawn openings for White, but I have nothing against other openings if they lead to speedy wins. Just send them along to me care of the editor of *TPUG Magazine* and, when space allows, we can publish the best of them. □

Sign on to CompuServe



For only $39.95* Cdn. get a complete CompuServe Starter Kit with detailed manuals and 1 hour of time. Please add $5.00 for shipping and handling. *U.S. send $28.95 + $5.00. Ontario Residents add 7% P.S.T.**

YES! TPUG, SEND ME THE KIT

SEND COUPON TO:

TPUG
101 Duncan Mill Rd.
Suite G-7
Don Mills, Ontario
Canada M3B 1Z3

NAME _____

ADDRESS _____

CITY _____ PROVINCE _____

POSTAL CODE _____

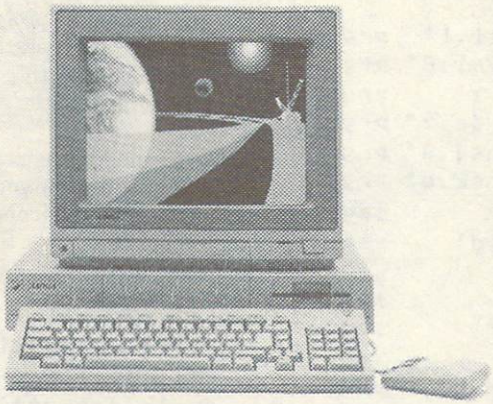
☐ I have enclosed my cheque for $39.95 + $5.00 shipping.

☐ Charge my Visa # _____ EXP. _____

☐ Charge my M/C # _____ EXP. _____

AMIGA

The future is here!

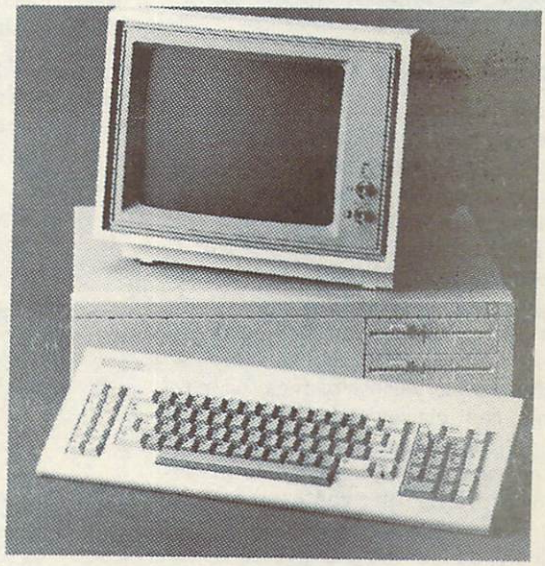


- Multi tasking
- 4096 colours
- Graphics speed faster than closest competitors
- Stereo music
- Voice synthesis
- Mouse driven
- Pull down menus
- IBM compatible too

NOW ON DISPLAY

COMMODORE PC10

FLEXIBLE, AFFORDABLE, AND IBM COMPATIBLE.



This is Commodore's remarkable new PC10. It's the flexible, affordable computer that can help businesses grow.
Among a wide range of functions the new PC10 runs the MS DOS operating system, the industry standard.
The PC10 comes standard with 256K memory, monochrome monitor, monitor card, two 360K 5¼" double density, double sided disk drives and on board serial and parallel interfaces.
It's modular in design. There are 5 available expansion slots which allows for a huge range of add-ons to be fitted.
A computer by any other name wouldn't be as good a value.
Or, more importantly, as good a computer.

$2995.00 CANADIAN

The complete guide to the 1541 operating system.
$25.95

$39.95

PaperClip
(the word processor)

$79.95
with SpellPack
$99.95

Boss Joystick

$23.95

Kraft Joystick

$17.95

Ribbons

1525	$12.95
1526	$14.95
4023	$14.95
Spirit 80	$14.95
801	$12.95
802	$14.95
8023	$14.95
4022	$12.95
MX80	9.95
RX80	9.95

MACHINE LANGUAGE FOR BEGINNERS BY JIM BUTTERFIELD $20.95

ELECTRONICS 2001 LTD.

Name _____

Address _____

_____

☐ Check or money order enclosed ☐ Visa ☐MasterCharge ☐ American Express

Acct.# _____ Exp. Date _____ Signature _____

ELECTRONICS 2001 LTD.

5529 Yonge Street, Willowdale, Ontario M2N 5S3
Tel: (416) 223-8400

Note: All prices in Canadian Funds. Phone and mail orders welcome.
Ontario residents add 7% sales tax.
Add 5% for shipping (minimum charge $2.00)

10% discount for TPUG members on regular priced software, accessories and magazines.

Library Additions

C-64 Disk (C)TL

Presented by Mike Donegan

This month's disk features a **Petspeed** version of **crossword 7/ml.c**, an educational program that allows you to generate your own crossword puzzles. Associated files include: **crossword 7.z** (the BASIC version of **crossword 7/ml.p**), **crossword 7 in.z** (the instruction program for **crossword 7**), **crossword ins1.w** and **crossword ins2.w** (**WordPro** files containing the instructions). Also included is a wide variety of files for outputting the crosswords to various printers (the filenames for these are prefixed with **pf**). If you need to make your own printer file, you can use the utility **cross.pf.maker.z** (which is also supplied), and for which instructions are contained in the **WordPro** files **cross.pf.ins 1.w** through **cross.pf.ins 5.w**.

Also on this disk is a freeware version of **Turtle Pilot**, a language designed to help educators write computerized lessons and text. It includes graphics, music and sound effect commands. The language is easy to learn, and allows complete control of screen colours, graphics modes and sprites. It also allows you to mix Pilot and BASIC in the same program.

The **list-me** file for this disk contains a directory of the commands available in this implementation of Pilot, and example programs are contained in the files **example1** through **example2**. These programs are from the **Turtle Pilot 64** reference manual (see the program for details on where to get this documentation).

Finally, this disk also contains two new programs from Jim Butterfield. One is a completely redone version of **Supermon**, called **Supermon + 64**. This one emulates almost exactly the built-in monitor of the C-128 running in C-128 mode. Thanks to Jim, C-128 users will not have to deal with two different monitors when they switch sides on their machines. Instructions for the new version will be found in the file **supermon + inst**.

Jim's other new program is called **list all**. It will produce a formatted listing to screen, printer or disk of any Commodore BASIC program (from any of the 65xx-based machines), and tell you which machines the program will run on.

```
0 c-64 disk (c)tl
26    "list-me (c)tl.l"   prg
91    "crossword 7/ml.6"  prg
81    "crossword 7.z"     prg
41    "crossword 7 in.z"  prg
23    "crossword ins1.w"  prg
15    "crossword ins2.w"  prg
2     "pf-mx80 i.d"       seq
2     "pf-mx80 iii.d"     seq
1     "pf-1526.d"         seq
1     "pf-2022.d"         seq
2     "pf-gemini.d"       seq
1     "pf-8023.d"         seq
2     "pf-prowriter.d"    seq
2     "pf-rx80.d"         seq
1     "pf-mps 801.d"      seq
3     "pf-mt160/180.d"    seq
2     "pf-teo px80.d"     seq
1     "pf-4023.d"         seq
2     "pf-spirit 80.d"    seq
2     "pf-6400 asc.d"     seq
68    "cross.pf.maker.z"  prg
27    "cross.pf.ins 1.w"  prg
32    "cross.pf.ins 2.w"  prg
31    "cross.pf.ins 3.w"  prg
31    "cross.pf.ins 4.w"  prg
24    "cross.pf.ins 5.w"  prg
24    "turtle pilot"      prg
1     "example1"          prg
5     "example2"          prg
2     "example3"          prg
1     "example4"          prg
2     "example5"          prg
3     "example6"          prg
3     "example7"          prg
29    "supermon+ inst"    prg
13    "supermon+64"       prg
24    "list all"          prg
```

PET Disk (P)TL

Presented by Mike Donegan

This month's disk features a **Petspeed** version of **crossword 7/ml.c**, an educational program that allows you to generate your own crossword puzzles. This is the same program that appears on this month's C-64 disk: for details consult the C-64 description above.

Also on this disk is **diskutil.8** — a disk utility similar to **disk doctor**. It is set up to work with the 4040 disk drive. Briefly, it enables you to look at disk files and blocks (hex or ascii format), and to edit any block on a disk. Instructions for this program are contained in **help diskutil.8**; and the **help use help.8** program will give you instructions on generating such help files yourself. You do this by inserting **DATA** statements into the program **help (empty).p**.

Finally this month, we have a freeware pacakage that will allow you to calculate taxes on AT&T shareholders. The package consists of two files: **divest.8 business** and **divest inst.8 instructions**.

```
0 pet disk (p)tl
20    "list-me (p)tl.l"   prg
73    "diskutil.8"        prg
18    "help diskutil.8"   prg
16    "help use help.8"   prg
7     "help (empty).p"    prg
93    "crossword 7/ml.p"  prg
81    "crossword 7.z"     prg
41    "crossword 7 in.z"  prg
23    "crossword ins1.w"  prg
15    "crossword ins2.w"  prg
2     "pf-mx80 i.d"       seq
2     "pf-mx80 iii.d"     seq
1     "pf-1526.d"         seq
1     "pf-2022.d"         seq
2     "pf-gemini.d"       seq
1     "pf-8023.d"         seq
2     "pf-prowriter.d"    seq
2     "pf-rx80.d"         seq
1     "pf-mps 801.d"      seq
3     "pf-mt160/180.d"    seq
2     "pf-teo px80.d"     seq
1     "pf-4023.d"         seq
2     "pf-spirit 80.d"    seq
2     "pf-6400 asc.d"     seq
68    "cross.pf.maker.z"  prg
27    "cross.pf.ins 1.w"  prg
32    "cross.pf.ins 2.w"  prg
31    "cross.pf.ins 3.w"  prg
31    "cross.pf.ins 4.w"  prg
24    "cross.pf.ins 5.w"  prg
25    "divest.8"          prg
7     "divest inst.8"     prg
```

VIC Disk (V)TK

Presented by Richard Best

Games: **Slalom**, a down-hill ski game/demo; **Torpilleur**, **Bombs Away** in French; **Spider Rescue**, like **Berzerk**;

SOFTWARE ORDER FORM

TORONTO PET USERS GROUP

101 Duncan Mill Road, Suite G-7
Don Mills, Ontario
M3B 1Z3

416-445-4524 416-445-4524

Name _____ **Membership No.** _____

Street Address _____

City/Town/P.O. _____

Province/State _____

Postal/ZIP Code _____ Telephone _____

DISKS

To order club disks by mail, send $10.00 for each 4040/2031/1540/1541 disk (4040 format), and $12.00 for each 8050/8250 disk (8050 format). We do honour purchase orders from school boards.

These disks are for use with a _____ computer and a _____ disk drive.

Please send me the following:

3 Letter/No. Code	Description	4040 or 8050 Format	Price

Total $ _____ .00

TAPES

To order VIC 20 or Commodore 64 library tapes, send $6.00 for each tape.

To order PET/CBM or Commodore Educational Software, send $10.00 for each tape.

These tapes are for use with a _____ computer and a datasette.

If for a PET computer, what model _____ - BASIC- 1.0( ), 2.0( ), 4.0( )?

3 Letter/No. Code	Description	Price

Total $ _____ .00

The prices indicated include postage and handling as well as Ontario Provincial Sales Tax (if applicable).

☐ Cheque/money order enclosed (payable to TPUG) ☐ Visa/Mastercard # _____

Expiry date _____ Signature _____

Caves of Ceti, similar to **Cobra**, with horizontal scrolling; and **Othello**, a perennial board-game favourite.

Utilities: **RDB Char Gen**, for custom characters; **System**, to list BASIC keywords; **Pause Key**; and **Supermon+**.

Productivity: **Simple Writer**, a minimal word processor; and **Cass Label 1 & 2**, for printing cassette inserts with titles (for both graphics and non-graphics printers).

Educational: **Planets**, to plot planet positions over time; **French 5K & 8K; a reissue that will work with tape or disk; Speedread**; and **Binary-fax, Hex-fax, Memory-fax** and **Software-fax**, a series of tutorials on computer basics.

Demos: **Enterprise**, for you Trekkies; **Gemini Demo**, for your Gemini printer; **Sound Story**, an illustrated fairy tale; and **Square Demos** 1 through 4 for SX.

A couple of notes: First, the instructions provided for **Supermon+** are set up for the C-64. Perhaps some enterprising member would like to convert them to VIC 20 format. And second, the stock of good programs for future library disks is becoming sadly depleted. Send in your stuff!

```
0 vic-20 disK (v)tK
10   "list-me (v)t-K/1" prg
12   "list-me/2"        prg
7    "planets.v"        prg
10   "rdb char gen.ins" prg
11   "rdb char gen.v5K" prg
4    "slalom"           prg
7    "simple writer.v"  prg
11   "torpilleur.v5K"   prg
5    "spider rescue.v5" prg
8    "spi2"             prg
2    "system.v"         prg
9    "enterprise.v5K"   prg
6    "gemini demo"      prg
7    "caves of ceti.v5" prg
9    "caves main"       prg
10   "cass label.v"     prg
10   "cass label.v2"    prg
10   "french drill.v5K" prg
12   "f"                prg
7    "sound story.v"    prg
7    "speed read.v"     prg
14   "binary-fax.v"     prg
14   "hex-fax.v"        prg
15   "memory-fax.v"     prg
14   "software-fax.v"   prg
2    "pause key.v"      prg
11   "othello.v5K"      prg
9    "french drill.v8K" prg
12   "f8"               prg
4    "square demo.vsx"  prg
4    "square #2.vsx"    prg
4    "square #3.vsx"    prg
2    "square #4.vsx"    prg
29   "supermon+.c"      prg
13   "supermon+.v12K"   prg
```

SuperPET Disks (S)TK and (S)TY

Descriptions of this month's SuperPET disks were unfortunately not available at press time. The directories appear below.

```
0 superpet disK (s)ty
16   "calc"             prg
12   "calcs"            prg
11   "which+calc:e"     seq
73   "DEVCALC"          prg
15   "devcalc+instr:e"  seq
8    "mfor+patch3:bp"   seq
7    "mbasic+patch3:bp" seq
9    "notes+patches:e"  seq
66   "memap+titles:e"   seq
66   "memap+address:e"  seq
11   "random+text:bp"   seq
24   "copy/Kill:men"    prg
22   "copy/Kill:ein"    seq
5    "clip:men"         prg
8    "clip+instr:e"     seq
7    "os9/pet.asm"      seq
11   "arrow.asm"        seq
1    "arrow.cmd"        seq
1    "arrow.lst"        seq
2    "arrow.b09"        seq
1    "arrow.map"        seq
1    "arrow.mod"        prg
1    "arrow:men"        prg
38   "format"           prg
71   "Format+info:e"    seq

0 superpet disK (s)tx
13   "describe.sep/85"  seq
50   "BEDIT"            prg
117  "bedit+instr:e"    seq
96   "bedit+tut:e"      seq
80   "batch+tut:e"      seq
30   "change+tut:e"     seq
17   "fixed+instr:e"    seq
14   "dos+commands:e"   seq
66   "BEDCALC"          prg
71   "calc+instr:e"     seq
94   "calc+tut:e"       seq
```

COMAL Library Update

Presented by Victor Gough

The second series of 25 COMAL library disks is now complete. The current library comprises 44 disks in seven series. Due to the increase in the size of the library, the disk header conventions have been elaborated. In addition to the usual three-character disk code and the COMAL-version identifier on the header, a machine-type code has been added, to allow for machines other than the C-64 to be identified. If no machine-type code is present, then the disk is assumed to run on the C-64. (K)L1 is an example of a COMAL disk that does not run on the C-64.

(K)Z8 and (K)Z9 are an expanded version on two disks of the tutorial disk (K)Z1. If this is your first look at COMAL, try either Z8 and Z9, or Z1.

Most of the books on COMAL in the book series are available through your local bookstore or the COMAL USERS GROUP, USA.

(K)L1 contains the new upgraded version for the PET/CBM computer line which replaces the older 0.12 COMAL. This has the same kernal as the 0.14 64 COMAL, but of course does not have the C-64 sound and graphic packages.

(K)Tx COMAL today Disk Series

(K)T5 MIXED COMAL today #5
(K)T6 V0.14 COMAL today #6 side 1
(K)T7 V2.00 COMAL today #6 side 2
(K)T8 V0.14 COMAL today #7 side 1
(K)T9 V2.00 COMAL today #7 side 2
(K)TA V0.14 COMAL today #8 side 1
(K)TB V2.00 COMAL today #9 side 2

(K)Ux User Group Disk Series

(K)U6 V0.14 User Group #6
(K)U7 V0.14 User Group #7
(K)U8 V0.14 User Group #8

(K)Bx COMAL Book Series

(K)B7 V2.00 COMAL HANDBOOK (2nd ed.)
(K)B8 V0.14 GRAPHICS PRIMER
(K)B9 V2.00 COMAL 2.0 PACKAGES
(K)BA V2.00 COMAL 80 (Commodore)

(K)Zx Miscellaneous Series

(K)Z5 V2.00 cartridge demo #2
(K)Z6 V2.00 cartridge demo #3
(K)Z7 V2.00 cartridge demo #4
(K)Z8 V0.14 Bricks tutorial #1
(K)Z9 V0.14 Bricks tutorial #2
(K)ZA V0.14 Utilities #2 side 1
(K)ZB V0.14 Utilities #2 side 2

(K)Xx Best of COMAL Series

(K)X1 V0.14 best of disk COMAL #1

(K)Lx New Language Series

(K)L1 PET V0.14 COMAL for PET/CBM

(K)Sx Slide Show Series

(K)S1 V0.14 hi-res pictures #1
(K)S2 V0.14 hi-res pictures #2

Super-OS/9* Is Here

TPUG has implemented the popular 6809 operating system OS-9* on the SuperPET. Super-OS/9 greatly expands the software availability and the hardware capability of the SuperPET while preserving access to the Waterloo languages and programs.

The cost of Super-OS/9 to club members is $195 (Cdn) (plus $10 shipment/handling Ontario residents add 7% PST), which includes the cost of a hardware modification that will not affect the normal operation of your SuperPET, installation instructions and the operating system disks.

To obtain your copy please send your cheque or money order to:

TPUG
101 Duncan Mill Rd., Suite G-7
Don Mills, Ontario
Canada M3B 1Z3

What does Super-OS/9 offer?

- A true disk operating with features found on UNIX* systems and on the AMIGA.
- Multi-tasking and multi-user capability.
- Hierarchical directory structure with time/date stamps.
- Programmable file access privileges for increased security.

Extensive software is available for OS-9, most of which runs on Super-OS/9.

Super-OS/9 V1.1 includes an assembler, editor, symbolic debugger, communication software and terminal emulation package. Available languages include BASIC09, CIS COBOL, Fortran 77, Pascal, Lisp, C and others. Application software include wordprocessors, spelling checkers, data bases and spread sheet programs.

TPUG has acquired public domain software and will assist users in the conversion of commercial software to Commodore format.

Portability and Expandability

- Super-OS/9 programs will run on all OS-9 based computers (like the CoCo).
- Super-OS/9 will support standard disk drives (IBM format) and the Hi-res graphics board (700 x 300 dots).
- Super-OS/9 software is C compatible with OS-9 68k and AT & T Unix system V.

For further information call TPUG Inc. at (416) 445-4524, ask for Alan.

NOTE: If you own a 3 board SuperPET and wish to acquire Super-OS/9, please call TPUG before, ordering Super-OS/9, for info about a hardware fix to a design error in your SuperPET computer.

Super-OS/9 is a trade mark of TPUG and Avygdor Moise.
OS-9 is a trademark of Microware and Motorola.
UNIX is a trade mark of Bell Laboratories.

OS/9 Software NOW AVAILABLE!

Word Processor – Stylograph III

Mail merge, Spell checker. Menu driven, fast. Extensive features. Formats to screen as you work. . . . . US 149 Cdn $189

C Compiler

A full implementation of K & R standard C, with double precision variables.
. . . . . US $120 Cdn $166

BASIC09

A fully structured, sophisticated semi-compiled (I code) BASIC. See Byte magazine April 1984. . US $110 Cdn $152

FORTRAN 77

Full FORTRAN compiler . . . coming soon.
. . . . . US $120 Cdn $166

DynaStar, DynaForm, DynaSpell

A full screen (similar to microEDIT) editor, text formatter and spell checker.
. . . . . US $000 Cdn $000

Other: Pascal, DynaCalc, Database etc., available on demand.

To Order: Send a cheque or money order (add $10 for shipping/handling and 7% PST where applicable) to:

TPUG,
101 Duncan Mill Rd., Suite G-7,
Don Mills, Ontario, Canada M3B 1Z3

COMSPEC

Authorized Commodore dealer since 1978.
866 Wilson Ave., Downsview
(Between Dufferin & Keele)
(416) 633-5605

Call our 24 hour B.B.S.
633-0185

• Hardware • Software • Books •
• Accessories • Service • Rental •

Colonial Conquest

$44.95

Field of Fire

$49.95

Computer Ambush

$59.95

Kampfgruppe

$64.95

Operation Market Garden

$59.95

AMIGA

NEW AMIGA NOW ON DISPLAY

10% discount to TPUG members
on Software, books and accessories,
(excluding sale items)

TPUG Library available for copy

Phone orders only accepted
Visa / MasterCard / Amex

(416) 633-5605

NEW

PRO-TERM 64

INTELLIGENT TERMINAL PACKAGE

© 1984 E. ANDERSON and G. FARMANER

Now with VT100 emulations
A full screen editor in the buffer
40/80 columns in software
X modem protocol
Steve Punter's new C1 protocol
40/80 column display
Informative status line
Extensive Auto-Dial/Auto-logon commands
Upload/download:
(Punter) (X/ON X/OFF) (.IMG FILES)
Complete Disk support, DOS commands, dual drive, or two single drives
Complete Printer support
Transparent stream to Printer
Programmable Function Keys
Remote-terminal modes, (Mini BBS), (Visual phone answer)
Supports VT/52 and Televideo 910/920 terminal emulations

On disk ONLY $46.95

KING MICROWARE Ltd.

Suite 210
5950 Côtes des neiges
Montreal, Quebec H3S 1Z6
Canada

C.O.D., money order or cheque

Dealer and distributor inquiries invited.

Write for our FREE catalog.
Send us your programs. We publish software.

C-64 & C-128* OWNERS:

Now YOU too can get the "Software Edge"!

Stop playing "Hide-and-Seek" with your Disk software! Let your Commodore 64™ "memory" do all the work for you with **MASTERDISK©**. Easy Manual and Program Menu-Screens make everyone a "pro" in 1 day!

Here's a small "PEEK" at SOME of the FULLY-INTEGRATED POWER of PRODUCTIVITY at your fingertips. . .

- a SORTED Master Catalog of ALL your Disks (even those un-listable "protected" Directories: Names, ID's, File info, etc.)
- Search + List any File/Disk/ID in seconds, automatically, in any of SEVEN WAYS.
- Transfer ML-Programs or Files from Disk to Disk (CATEGORIZE your software)
- Stop disaster before it strikes: change duplicate ID's fast, without total re-copy.
- Restore accidentally-scratched files in a flash, with VERIFICATION that they're OK.
- Works with 1 or 2 Single Disk Drives (1541, MSD, etc.) with or without Printer (Commodore or ASCII).
- Handy-Reference Card for HEX/ASCII/BINARY-TO-DECIMAL VALUE TRANSLATION + FORMULAS, included FREE.
- All this + MUCH MORE, for just $34.95. (U.S.A.), $45.95 (Canada)

NEW!! - MASTERDUAL©, "Son of MASTERDISK©" . . .
same power, but with Dual Drive Speed!

- Same Catalog Features as MASTERDISK©
- Same Disk repair functions, too
- For all C-64 & C-128* systems with Dual Disk Drives (4040, MSD, etc.)
- Utilizes all Dual Drive functions for maximum speed and ease of use
- FREE Handy Reference Card for HEX/ASCII/Binary to Decimal Translation + Formulas
- Experience FULLY-INTEGRATED POWER for just $29.95 (U.S.A.), $39.95 (Canada)

***MASTERDISK© and MASTERDUAL©** are also compatible with the NEW C-128 (in C-64 Mode)!

We ship within 24 hours! **Send Money Order or Check to:**

INTEGRATED-SOFTWARE SYSTEMS
P.O. Box 1801, Ames, IA 50010
VISA/M.C./C.O.D., Call **(515) 233-2992**
9 a.m. - 9 p.m. (Central Time) Mon.-Fri.
Add $3 Shipping/plus $2 Foreign, C.O.D.
IA Residents add 4% Sales Tax.

IN CANADA: Cricket Distribution Co.,
P.O. Box 953, Cobourg,
Ontario, Canada K9A 4W4
VISA/M.C./C.O.D., Call **(416) 372-3692**
9 a.m. - 6 p.m. (Eastern Time) Mon.-Fri.
Add $3 Shipping/plus $2 C.O.D.
Ontario Residents add 7% Sales Tax

Whether you have tens or hundreds of disks, now YOU can be the MASTER of your TOTAL Software Library!

Switch

to the highest quality, lowest priced brand name diskettes available in Canada.

Q. Who has switched?
A. Professional and educational institutions, national accounting firms, major engineering firms, software houses, computer retailers and the Toronto Pet Users Group.

Limited time offer

SSDD - $15.00/box
DSDD - $20.00/box
(Minimum purchase - 5. Delivery extra.)

Programmers Guild Products Ltd., 255 ManuLife Place, 10180 - 101 St.,
Edmonton, Alberta T5J 3S4 Phone: (403) 428-6229

VISA MasterCard

Reviews

C Power
from Pro-Line Software
C language compiler for Commodore 64

Review by C.G. Conville

"C?" you ask. Not *the* C? Yes, a real full-fledged alternative to BASIC is now available for the Commodore 64. This new programming package from Pro-Line Software attempts both to fill the void left by BASIC's problems and to satisfy the needs of more experienced programmers. But first, let's answer a few questions.

What is C?
C is a (usually) compiled language that was developed for the Unix operating system on the DEC PDP-11 minicomputer in the 1970s. It's 'official' definition can be found in the book *The C Programming Language* by Kernighan and Ritchie, commonly referred to as K&R.

Okay, then, but what's it like?
C is an offshoot of B (makes sense, right?). It is a 'structured' language, with resemblances to Pascal and ALGOL. Unlike BASIC, it lets subroutines (referred to as procedures and functions) pass parameters, and permits 'local' variables (variables totally separate from the main program or mainline). As well, C provides for macros and object-code libraries.

Great! But what are you talking about?
The main point here is that you can compile (if you want) just a few subroutines, and save them to disk for future use. In other words, you can program your task in small pieces and then, when done, bring them all together in a finished program. This allows for your brother Phil to work on the input/output routines, and your sister Andrea to come up with the calculation routines, while you wrap everything up together with a mainline.

Understood. But why should I want to use C on my C-64?
Three reasons: It executes faster than BASIC; structured programming will help you in writing the program; and chances are good that your program written in C can be transported to other computers (micros and minis) with little or no modification. This is because C is a very standardized language.

So C is compiled? I'm used to interpreters. What is involved with a compiled language?
As you may know, with an interpreted language (like BASIC), your program is interpreted as it is executed, instruction by instruction. This makes program development easy, but results in slow execution. With a compiled language, your program is typed in and modified first using a program called an editor. This part of your program is called the source file. Then, a compiling program reads this source file and translates it into machine code (though some compilers translate into an intermediate code called 'p-code', which is not as fast as machine language). It may seem like more work using a compiled language than an interpreted one, but compiled code has one big advantage over interpreted code: it's *much* faster.

Now let's turn to the **C-Power** package itself.

What's included

The **C Power** package consists of a double-sided disk, an attractive (and bulky!) manual, a stapled booklet of additional documentation, and two cards used for obtaining a backup and getting update information from Pro-Line. The manual is actually a proper book — *C Primer Plus*, by Waite, Prata and Martin. This looks like a perfect 'first book' for the starting C programmer. It is written in a friendly style, with frequent comic-like illustrations, summaries and exercises/projects. A nice added touch is a pull-out-and-fold quick reference card to the C language.

The additional documentation booklet covers the basics: how to load in various parts of the **C Power** package, the commands and options for each program, and a brief list of anomalies (departures from the K&R standard). The last pages list the various library functions included with **C Power**, along with descriptions and brief examples. These routines can be called from any C program, and provide for useful functions such as file i/o, string-handling and advanced mathematical functions.

Finally, the disk itself! The double-sided disk includes a fancy command-interpreter called a shell, two editors, the compiler/translator, a linker, some example programs, and numerous files for the function library mentioned above.

The Shell

The Shell is a program that provides an environment for developing and compiling your programs. It allows you to call up various features of the **C Power** package using simple one-line commands. For example, to edit the file **felgercarb.c**, it suffices to say:

ed felgercarb.c

The shell provides similar commands for compiling and linking your programs. Experienced Unix users (who are already familiar with the shell concept) will recognize the similarity between commands on such systems and those in **C Power**'s shell. If you choose to compile your programs so that they are run under the shell (this is a linker option), then they are executed by simply typing the program name. In effect, your programs add new commands to the shell.

Like Unix shells, **C Power**'s shell supports i/o redirection. This allows the user to divert input or output operations from the keyboard and display (called the standard_input and standard_ouput in Unix terms) from or to disk files. For example, if a program asks for ten phone numbers as input, then every time you test it you normally must type in ten numbers. Using redirection, you can make the shell read the numbers from a disk file (created with an editor), eliminating the need for retyping. Output redirection could be used to save the output from a program's execution for future use. Input and output redirection are indicated by use of **<filename1** and **>filename2** in the shell's command line after the command/program name. Diverting output to the printer is performed by using **>>** (note to Unix users: this is *not* an append).

Looking at the Shell again from a Commodore perspective, it also provides commands for getting disk directories and sending DOS commands via a 'wedge'-like command. And if you are fortunate enough to have two disk drives (or one double drive), you can designate device/drive numbers for your work disk and system disk. Thus you can edit/compile/link your program without ever having to swap disks. This is a real convenience. The commands used for changing device numbers (the **sys** and **work** commands) do not, however, actually change them in the drives. These commands only tell **C**

Power which disk is which. You will need to change the disk's device numbers beforehand (programs to do this are available in the public domain).

The Editors

No programming environment is complete without an adequate method of creating and modifying your program's source-code. **C Power** provides two editors for this purpose, called **ed** and **ced**. **ed** is a full-screen editor, with an additional command mode for non-screen functions (like reading or writing a file from/to disk, getting disk directories, a disk wedge, and quitting the editor). It also allows the use of buffers, which can be used to hold material for editing, and can be manipulated independently. A typical situation where buffers could prove useful is where a programmer is working on a mainline to a program in one buffer, and keeps a set of associated functions together in a separate buffer. You are thus able to work on one thing without having to look at the other constantly.

However, by far the most important mode of the editor is the full-screen mode. This is where you will be, 90 per cent of the time. Here, you type in your text in free format, inserting and deleting much as you would with a word processor. If your line goes over 40 columns, the screen scrolls horizontally as you continue typing. Through some experimenting, I determined that the maximum line length is 240 characters (though the documentation doesn't mention this). This is easily more than you are ever going to need, but it's nice to know it's there. Cursor movement is accomplished, not surprisingly, by using the cursor keys, although they may appear to behave a little strangely. For example, instead of simply cursoring down, the cursor is left on the first non-blank character of the line. As well, you can advance forwards and backwards a screen at a time using the page-down and page-up commands. Full search and replace functions are available, along with a 'cut-and-paste' feature. And since C uses special characters (such as braces and vertical bars) in programs, the editor provides these also, using special key combinations listed in the editor documentation.

ced acts identically to the regular editor, except that it will, on command, perform a rudimentary syntax check on whatever source-file is in memory. I say 'rudimentary' because the documentation does not indicate how extensive the error-checking is. It does, however, check for problems like unclosed parentheses, quotes and braces, which are the sources of errors in most cases. I would strongly advise using this particular editor for most of your program development. There's nothing more aggravating then having to recompile a program several times simply because of typing errors (like **ulse** instead of **else**).

The Compiler

Any program that is to be compiled must have a filename ending with '.c'. This is not a problem at all, since it makes source files easier to spot on a disk directory. The shell command **cc <filename>** starts compilation. Note that quotes are not needed. If you are using a single drive, the compiler will ask you to swap disks at certain points. As your program is compiling, it is listed to the screen. If an error is found, this listing (and the compile) pauses until you press a key to continue. This prevents any possible errors from scrolling off the screen while you are out of the room, for example. When the compile is finished, the object file will be on disk with a '.o' suffix.

The Linker

Since C programs can be written in several parts, and these parts can be written separately, it becomes necessary to join the parts together, or 'link' them, to create the final runnable program file. This is the purpose of the shell command **link**. The linker prompts you to type in the names of your object files on the work disk, and reads them in. Hitting **RETURN** in response to the prompt will give you a listing of unresolved references (the parts it still needs before it creates the final program). Usually, these references will be calls to the function libraries. For example, since most programs require input and/or output, the library functions **scanf** and **printf** will probably be listed as external references. **C Power** provides three standard libraries, called **stdlib.l**, **syslib.l** and **mathlib.l**. Most programs will probably need the first two in the linking stage. They are found on the system library disk. Finally, when all references have been found by the linker, it will ask you for the filename to write to. This file will be what all this work is for: the final product. You may specify whether the linked program is to run inside the shell (the default), or as a regular program file to be run from BASIC.

The Libraries

As mentioned above, **C Power** provides three function libraries. The library **syslib** is the largest (and, when looking at a directory, the one with the most incomprehensible filenames). Since most of the filenames start with 'c$', I think that this library is more for object-code requirements than for callable functions like **scanf**. The **stdlib** library contains the input/ output and string functions, along with a few handy ones that perform tasks like allowing you to call a ROM routine, to set the top-of-memory pointers used by **C Power**, or to sort the contents of an array. The library **mathlib** contains the standard logarithmic and trigonometric functions, as well as hyperbolics, square root, powers and various rounding functions. In other words, these libraries are very complete.

Other Goodies

The **C Power** package also provides several shell utilities (taken from *Software Tools In Pascal* and, of course, K&R) to aid in program development. The first of these is **find**. This command will check the standard input for a given pattern of characters. The next utility is called **sort**. As its name implies, this command will sort the standard input alphabetically into the standard output. An option is provided to allow for numerical order as well. The third utility, **wfreq**, has comparatively limited applications: it counts the number of occurrences of each word in the standard input.

Better Goodies

Although those utilities may have their uses, the most useful one provided is **format** which, as its name implies, is a text formatter. Anyone who has used a word-processing program (**PaperClip**, **WordPro**, **Speedscript**, and so on) will be familiar with the idea of a file containing your text along with commands to perform tasks such as centering or indenting. Although not as powerful as a dedicated word-processor, **format** does provide the essentials: margin control, centering, indenting, paging, and headers and footers. Once you've prepared your text file with **ed** (or **ced**), you use the **format** command, specifying the file(s) to use for formatting. The formatted text is sent to the standard output, so it can be redirected if desired. Useful as **format** may be, it may be too much work simply to get nice program listings. The **print** command takes care of this: it prints a listing with margins at the top and bottom of each page, along with a header at the top of each page. This is probably handier than the shell's **pr** command, which does no formatting.

Benchmarking

After playing around with the package for a while, I decided to see how well it performs in comparison with BASIC. To do this, I decided to implement the well-known Sieve of Eratosthenes, which is commonly used by *BYTE* and other magazines for benchmarking purposes. This is a technique used to find prime numbers. Although I don't have space here to give details, the outcome was heavily in favour of **C Power**, especially when the program was written to use integer rather than floating point variables. Compiling the BASIC version with the **Blitz!** compiler produced a program that was somewhat faster than **C Power**'s for floating point operation; when integer versions were used, **C Power** retained a considerable advantage.

Problems

Since I had ordered mine early, I received one of the earlier copies of **C Power** (version 1.0). It had problems. To be fair, the problems weren't serious and could be worked around, but they were still annoying. Calling Pro-Line and talking to a very helpful man named Stew Martin got them aware of the problems (most had already been corrected) and arrangements were made to get an updated version (my current one, version 2.2; further updates are being made). This level of support should be applauded.

However . . .

There were also differences in documentation between the two versions. For example, the original booklet listed the **run** command in the shell: it was curiously missing in the second version of the booklet (it turns out that this command isn't needed anymore). What bothered me in the original documentation was the statement that command-line arguments were not supported. The second version of the documentation made no mention of them (I did some playing around: they *are* supported, and work well). This sort of haziness could be avoided by a clearer booklet. I have a few other gripes about it: it's too small physically (the paper size is too small to fit in a standard binder), it's a little disorganized, lacks a proper index, is too brief, and it doesn't document all of the utilities. My biggest gripe concerns the part of the booklet that documents the editor. All it does is give a command list; a few examples would have been appreciated (especially concerning the find-and-replace commands). I don't want to sound *too* negative: the booklet does have all the needed information; it's just that I wish it was written with the user (especially the beginner) in mind.

Since most of your development time will probably be in an editor, it makes sense that it be as user-friendly as possible. For the most part this is the case with the **C Power** editors. However, when you edit a file you created previously, and then attempt to save it, you get the DOS error message indicating that the file already exists. This is more than a little annoying, since you then either have to scratch the old version (using the editor's built-in disk wedge) or take your chances with the infamous save-with-replace technique. Maybe I'm being a little picky, but it would have been nice to have the editor ask me if I wanted to overwrite the old version (by scratching and then saving). After all, a good program is written with the user in mind.

Another little thing is the matter of printer support. The big problem is that the documentation doesn't indicate how much support there is! Since I don't use a Commodore printer, I can't really experiment, but I'll make an educated guess: I don't think that the printer commands will print any of C's special characters on Commodore printers. Again, I may be completely wrong on this (how can I tell?) but the documentation should have made some note of **C Power**'s degree of printer support.

My biggest complaints concern the diskette itself. First and foremost, the compiler is copy-protected. True, the drive head doesn't bang, and the people at Pro-Line say that it will work on just about any drive (1541, MSD, 2031, 4040), but I still would like to be able to make a backup. Their solution is to offer you one for 20 dollars. I don't think much of that solution. It might be fine for game programs, but something as important as a compiler should not be copy-protected this way. I wish they had opted for dongle-protection instead, as Batteries Included has done for **PaperClip**. This allows you to make as many copies of the program as you want, but still provides excellent protection.

My other complaint about the diskette is that they've used both sides, one for the compiler and shell package, the other for the library modules. Though it is still a somewhat controversial question (as was the save-with-replace bug, until recently), I have heard many horror stories from computer users who double-side their disks: disk errors, files disappearing, and so on. Fortunately, the only file that is copy-protected on the **C Power** disk is the compiler itself. Thus you can copy everything else onto separate disks if you want (and you should, if only for peace of mind). Double-sided disks aren't a good idea: I wish companies would avoid them.

Definite Pluses

As mentioned before, the major part of the documentation is the book *C Primer Plus*. After reading through it a few times, I've reached the conclusion that Pro-Line made an excellent choice in including it: it is geared towards the beginner (unlike K&R), although some prior programming experience is assumed. The 'official' C reference (K&R) might scare you off; this one won't.

C Power supports just about every disk drive possible. For me, this is a real plus, since I use my older 2031 with a BusCard for my work disk, and a 1541 for the system disk, along with my Brother printer for listings. **C Power** can handle this odd assortment of devices (IEEE, serial and Centronics parallel) without problems.

The compiler produces good, fast code. This is what it all really comes down to, isn't it?

With the exception of bit-fields, a seldom-used feature of C, **C Power** provides standard K&R C, thus providing the programmer with the full power of the language. System-dependent features (like length of variable names and size of integers) are well-documented and reasonable.

My Wish List

As complete as **C Power** is, there are a few things that I would like to see (are you listening Pro-Line?):

- a standard-sized documentation booklet;
- a quick-reference card for the editor;
- a key-based protection system;
- the **%e** format in the library function **printf**;
- functions to access the system clock;
- separate disks (instead of double-siding);
- some description of printer support;
- better explanation of how to use libraries.

C Power is an extremely useful tool, offering a standard and powerful language to Commodore 64 programmers. It is well-packaged, well-documented and well-priced ($129.95 Cdn.) and does what it claims. **C Power** is recommended. □

Blazing Paddles
from Baudville
Art program
for the Commodore 64

Review by Mike Martin

Blazing Paddles is a versatile art program from Baudville. It allows the use of a light pen, trackball, joystick, touch tablet or paddles as input devices; but only one of these at a time. To change implements, you must power down and load the program again — a two and a half minute process, which the EPYX Fast Load Cartridge does not seem to speed up.

The menu features such functions as brush choice, colour selection, sketch, dots, lines, ovals, boxes, various text fonts, cut and paste window, library of shapes, fill, spray and zoom. The package supports Gemini 10X, Epson 80 and 100, and Commodore 1525 printers, with Cardco and Buscard interfaces.

The strong point of this package is the availability of 'shapes and fonts' library disks, which include outline drawings of animals, buildings, furniture, electronic symbols, trees and architectural elements. Three libraries are available, including a general disk, a science-medical oriented disk, and one for kids. Each includes various font styles that may be loaded. These libraries may also be used with **Animation Station**, from Suncom.

The package itself is a translation of an Apple program. The documentation is oriented toward the Apple computer. In the three 'shapes and fonts' libraries (available at additional cost), the manual includes a sheet with corrections for the Commodore version. The sheet states that not all of the drawings shown in the manual are included, and the drawings may vary from the pictures in the manual. In the Apple version, a file is loaded containing about twenty drawings of one category, and you 'page' through them. In the Commodore version, there are four or five files for each category, and no indication of which drawings are in each file. It is therefore necessary to access the disk up to five times, in order to discover which drawings are in which file; and which are missing. If you do this, you will want to make notes in the booklet for future reference.

The program is a bit slow to use with a joystick, as the cursor speed is not adjustable. All this disk access involves moving the cursor to the 'disk' block in the upper right corner of the menu, pressing the joystick fire button, using cursor controls to highlight a command, pressing the **RETURN** key to load a file, **RETURN** to go back to the picture, moving the cursor back up to the 'shapes' box at the top right corner of the menu, **RETURN**, moving the cursor to the commands at the bottom left corner of the screen to page through the few drawings in that file, moving the cursor back to the right bottom corner of the screen to select 'menu', and back up to the 'disk' block in the upper right corner of the screen to start the process again. This would be faster with a light pen. To leave a drawing, you have to remove the cursor from the screen (**f7**) and hit **f5** then **RETURN**. A single key, such as the back arrow, Commodore key or 'control' key would have made more sense.

The shapes are well done, but figuring out where they are in relation to the cursor is difficult. After placing them on the screen, moving the cursor and hitting the space bar moves the drawing or line of text to the new position. The space bar also acts as an 'undo' button, to remove the drawing or text entirely.

Colour can be selected by using the cursor; and two or three colours may be mixed, if desired. The 'spray' feature allows an even dot pattern similar to that of an airbrush to be sprayed on the screen. Going over the same area gives a more dense pattern. This is an excellent feature. In the 'line' and 'oval' modes, a 'rubberband'-style sketch appears on the screen, showing each move and change you make. Triangles can be plotted this way also. The 'zoom' feature shows half of the full-sized drawing, as well as a boxed, magnified version. You can draw, but only one pixel at a time; and, to erase, you must change the drawing colour to the background colour, drawing over your mistake.

A screen dump is included, although the results are uneven. The program adds a texture to each colour for printing, so a drawing may look terrible when printed. The printout is half a page, and prints right side up, so it could easily be used to print a letterhead at the top of a page. The screen dump only has one size: the full width of a page.

All things considered, the program is strong and useful, but slow to work with. There are so many different levels of quality to the package that I wonder if it was meant to be a 'crayons for kids' program that turned out to be excellent, or a 'professional art and illustration' program with serious lapses. □

Combat Leader
from SSI
Military strategy game
for Commodore 64

Review by Dave Dempster

Is your wrist tired from chase'em-and-gobble'em games? Your trigger finger sore and calloused from shoot'em-ups? Do you look for a game requiring intelligent thought and numerous tactical decisions, while offering continuous action with virtually infinite game options, as well as an exciting edge of uncertainty? Does SSI have a game for you!

Combat Command puts you in charge of a mixed arms force ready to do battle against a similar foe over a highly variable terrain. Your force can consist of a maximum of 59 units made up of any combination of:

- 3 tank platoons of 5 tanks each
- 2 carrier platoons of 4 carriers each
- 1 scout platoon of 4 vehicles each
- 2 infantry platoons comprising MG, AT, mortar and rifle sections

Terrain variables such as trees, hills, rough terrain and cover vary from 1-8, for a maximum of about 32768 different fields. Besides the six canned scenarios, you are offered the intriguing possibility of designing your own. Tanks and carriers can be 'designed' from armour, speed and gun capabilities, with 1939 to present-day systems listed — a total of seventy-two types. The speed of the scenario can be preset, so we slow thinkers don't get blown out of the game before we know we're in it. This game does not run by 'turn', like a board game, but is continuous. In other words, if you do nothing, the computer will move down the board and engage, with no time-outs.

The program first inquires about your selection — 'novice' is a good first choice, as you pit one platoon of tanks against a similar computer-directed unit. The game field comes up showing about a 40 by 24 area (the whole field is about 77 by 40), with your units shown on the terrain. You can scroll around the field, but enemy units are not displayed unless visible to one of your units, and may disappear again, using smoke or terrain cover, or by destroying your spotter. Your units communicate to you by message (seen on the bottom of the screen); and you send messages by typing

short commands such as: **A** (to A Platoon) **G** (go to present cursor position). You direct your units to move, provide patrols, inform them where to expect the enemy, when to open or cease fire; and you can also order them to mount or dismount (for carriers and infantry). Nothing to it, eh? Except that your units may panic and run, miss the target or...

Unlike computer chess games, (which don't merely beat me, but humiliate me too), I found I *could* beat the computer, particularly in the novice scenario, while learning to use my units. However, the addition of mortar, rifle and anti-tank sections with smoke, fire, and patrolling scouts is a real challenge. If you *really* think you're good, crank up the speed to 8, and try to keep up.

An interesting option lets you command a platoon consisting of a company-sized force, while the computer takes command of all the other units, including the oppositon. To get an idea of what I was up against, I set up the attack scenario, set speed at 8, and took command of a rifle section. It was very impressive.

After several friendly games with my C-64, I determined that the time was ripe to humiliate the beast. I constructed my 'own scenario'. I supplied myself with fast supertanks, bags of armour and a super-gun, while I gave the C-64 cardboard boxes on wheels, armed with popguns. I set the terrain up as flat — didn't want any slip-ups — and away I went...

I got a bit of a shock. I forgot to tell you how to win...

Depending on the scenario, points are allocated for damage done versus damage received. Point values depend to quite a degree on scenario balance. I happily swatted 10 of the C-64's boxes, and only accrued 10 points — so much for short cuts.

This game offers a fast-moving, fascinating and variable tactical opportunity. The small and concise manual will have you up and playing in fifteen minutes or so. Playing time varies from ten minutes to about an hour. The graphics are serviceable, but not remarkable. The sound of exploding shells and whistling mortars on top of the pop of small arms represents the total sound effects, apart from a very fierce intro tune. There is no air support, off-board heavy artillery or mine option. The effectiveness of the mortars is somewhat exaggerated, particularly against armour.

I highly recommend this game. It's not perfect, but it's the best of the thirty or so I've seen so far. □

Science and Engineering for the Commodore 64

by Raniear Bartel
from Abacus Software
343 pages

Review by Richard Goodson

One in an excellent series by Abacus Software, this book describes the use of the C-64 for solving problems in mathematics, science and engineering. Definitely not for light reading, the book assumes a strong background in mathematics and a knowledge of programming in BASIC. For the reader who fits this description, the book is a wealth of information, containing lots of useful programs.

The first two chapters deal with the use of BASIC. Problems created by numbers being rounded off and the lack of structured programming commands are discussed. The book explains the use of flow charts and structograms; and utilizing pseudo-code is also mentioned. Structograms are used to give a visual representation of blocks of structured programming. The author recommends the use of structured programming, and discusses how it can be implemented on the C-64. Following a section on variables, functions and operators is a section that describes how to trap errors, and how to input a mathematical function without interrupting the program.

Chapter 3 deals with input and output on the C-64, as well as with files. It explains the operation of a program to write, read and change records in a relative file, and the storage of vectors and matrices in files is lightly touched on.

Sort routines are dealt with in chapter 4. The bubble sort, linear sort, shell sort and quicksort are explained. A program to compare sort routines is given.

The remaining 232 pages are packed with descriptions of various problems, with programs that can solve them. This is where the going gets tough. For the non-scientist, most of the rest of the book will be practically meaningless. It is not easy to follow the explanations, which were obviously written on the assumption that the reader has a very thorough understanding of mathematics.

Chapter 5 contains programs dealing with zero point determination, differentiation, integration, linear regression, probability, Fourier analysis, differential equations, vector calculations, and matrix calculations. In chapter 6, there is a program to produce a relative file of the periodic table, plus programs covering pH calculations, titration, the gas laws for real and ideal gases, and quantum mechanics calculations of chemical bonding.

Physics problems are found in the next chapter. The first program is a three event timer controlled by the keyboard or external contact connected through the joystick ports. Other programs deal with the detection of faults in an underground cable, geometric optics and planetary orbits.

Biologists have to be content with a single program dealing with population dynamics and the predator-prey model of Volterra.

In chapter 10 there are programs dealing with heat transmission, pulley belt length calculations, and the analysis of complex electronic networks. The latter topic is a large one that takes up thirty-five pages and has four programs covering complex number conversions, complex impedance calculations, network current analysis, and node potential analysis.

The final chapter has some suggestions and equations that may be useful to someone interested in writing a CAD program, or a program to produce printed circuit board layouts.

This book would be great for the university science student, or a person who has a technical background, but the content is way beyond the grasp of most mere mortals. □

Online Guide

by Mike Cane
from Signet/New American Library
Softcover book
384 pages, $9.95

Review by Jim Strasma

This Commodore-specific extension to Cane's earlier book, *The Computer Phone Book*, helps owners of Commodore computers and modems successfully to communicate with the world via computer and telephone. Unlike some competing books, it omits lengthy discussion of the dozens of specialized words related to using a modem, such as parity, stop bits

and the like, and simply tells how to set various switches on Commodore 1600 and 1650 modems, so that they work most of the time. It also reviews equipment and terminal software packages directly usable by Commodore owners, giving very candid opinions of each (despite its modest claim that these are not reviews at all).

The heart of *Online Guide* is a conducted tour through the CompuServe Information Service, an information service housed in several large computers that home computerists may dial into and use for an hourly fee. CompuServe is well-known to Commodore owners for the support it provides for their computers. In the past, the Commodore-specific areas on CompuServe were managed by Commodore's Customer Support department; they are now run by TPUG.

Online Guide also provides brief introductions to two competing information services, Delphi and The Source; and a long section on Punter Bulletin Board Systems. These are local information services, running on Commodore 8032 or 64 computers, using telephone-answering software written by Steve Punter of Mississauga, Ontario. For several years, the Punter systems have been the primary telephone message system used by Commodore owners. They are also one of the primary ways Commodore owners exchange public domain programs and data files over the phone (along with CompuServe).

Online Guide's way of explaining these services is with extensive printer dumps of what one would see at various places within a particular system, with added comments and suggestions by the book's author. This approach will appeal to people who want something next to the computer to guide them as they hop from menu to menu within a particular information service. On the other hand, it is not well-suited to those who need a quick reference to a particular feature of a given service.

There is also the unavoidable problem of continuing change. Even since this book was written in 1984, some things have changed at CompuServe (the electronic mail system and the Commodore files system, for example). To cope with this problem, the author offers a monthly update service for the book, at a cost of 20 dollars per year (twice the cost of the book itself, but probably justified for those who really need the information).

One other feature of real but temporary value in the book is its discussion of actual public domain programs available for copying on CompuServe, and on some particular local Bulletin Board Systems around the country, when the book was written. Not all of these are Punter boards, so prepare to be briefly confused at times, when you call them.

On balance, *Online Guide* is detailed, as accurate as such a book can be, and good value. It can easily save you more than its own cost in long distance and access charges. Those who use a Commodore modem regularly will want a copy nearby at all times. □

Doodle Visits the Print Shop

from Software Link

Graphics utility for Commodore 64

Review by Edward K. Crossman

Doodle Visits the Print Shop *from Software Link, 283 Mamaroneck Avenue, White Plains, NY 10605. $25.95 US (disk).*

The picturesque title of this program may take you back to the days of the *See Spot Run* series of readers that are a hazy memory to most of us. But if you own a copy of the **Doodle** graphics program for your Commodore 64, and also have the very popular **The Print Shop** program, you can rejoice. Thanks to Andrew Viola and Software Link, there is now a way to integrate those marvellously creative images you produced with **Doodle** into **The Print Shop**. This means that you can take a **Doodle** picture and overlay any of **The Print Shop**'s fancy lettering fonts on your drawing. Then you can print this rather phenomenal union of images to your printer and save it to disk. Or, if you prefer, you can load your **Doodle** picture into **The Print Shop**'s screen memory and overlay your drawing with one of the many kaleidoscopic pictures **The Print Shop** is capable of creating.

The Graphic Editor mode in **The Print Shop** allows you to flex your imaginative muscles by creating your own graphic images using the keyboard, joystick or KoalaPad. Or you can call up one of the original **The Print Shop** graphics, such as a birthday cake, and modify it.

But suppose you prefer to create your own pictures with **Doodle**. Now you can load your picture into **The Print Shop** through the Graphic Editor mode, then combine it with **The Print Shop** graphics or save it to disk so that you can use it in a greeting card, for example. The advantage of taking this route is that in drawing your picture you can use many of the excellent **Doodle** drawing features that the Graphic Editor in **The Print Shop** lacks.

There's more good news. Software Link saw fit to make the street run both ways. Pictures that you've created with **The Print Shop** can be converted to **Doodle** pictures, and then modified with **Doodle**.

Finally, there is a 'retrieve colour' option that allows you to construct a colour picture with **Doodle**. That picture is then loaded into **The Print Shop** and you add a fancy title to it. The colour is temporarily lost at this point. Worry not, however. The 'retrieve colour' option in the **Doodle Visits the Print Shop** program allows you to restore the original colours. You now save this picture as a **Doodle** picture that can be displayed in glorious colour with the **Doodle** program.

All of this sounds a bit complicated, and initially it does take some practice to keep track of three or four disks that are being inserted or removed from the disk drive. Even more critical, though, is the necessity for careful planning. When a **Doodle** picture is overlaid with lettering from **The Print Shop**, much of the picture could be wiped out, depending on the size and quantity of the letters you use. This is not a disadvantage; it just means that some practice and forethought is necessary to produce a pleasing result. The main point is that Software Link discovered a gap between two graphics packages and figured out a way to perform a highly successful marriage. Let the honeymoon begin! □

Calendar of TPUG Events

Meeting Places

Brampton Chapter: Brampton Public Library, Downtown Branch, 1 block east of Highway 10 on Highway 7 (Queen St.), at 7:30 pm.

Business Chapter: Meeting place to be announced — consult the meetings line.

Central Chapter: Leaside High School, Bayview & Eglinton Aves. on the second Wednesday of the month, at 7:30 pm in the auditorium. For 'advanced' computerists.

COMAL Chapter: York Public Library, 1745 Eglinton Ave. W. (just east of Dufferin) on the last Thursday of the month, at 7:30 pm in the Story Hour Room (adjacent to the auditorium). Note: location may change in October — for latest information call 445-9040.

Commodore 64 Chapter: York Mills CI, 490 York Mills Rd. (east of Bayview) on the last Monday of the month, at 7:30 pm in the cafetorium.

Eastside Chapter: Dunbarton High School (go north on Whites Rd. from the traffic lights at Highway 2 and Whites Rd. to next traffic lights; turn left to parking lots) on the second Monday of the month, at 7:30 pm.

Communications Chapter: York Public Library, 1745 Eglinton Ave. W. (just east of Dufferin) on the first Wednesday of the month, at 7:00 pm in the Story Hour Room (adjacent to the auditorium). Note: location may change in October — for latest information call 445-9040.

Hardware Chapter: York Public Library, 1745 Eglinton Ave. W. (just east of Dufferin) on the second Tuesday of the month, at 7:30 pm in the Story Hour Room (adjacent to the auditorium).

New Users Chapter: TPUG Office, 101 Duncan Mill Rd., Suite G-7, Don Mills, on the dates listed below, at 7 pm.

SuperPET Chapter: York University, Petrie Science Building (check in room 340). Use north door of Petrie to access building. On the third Wednesday of the month, at 7:30 pm.

VIC 20 Chapter: York Public Library, 1745 Eglinton Ave. W. (just east of Dufferin) on the first Tuesday of the month, at 7:30 pm in the auditorium.

Westside Chapter: Clarkson Secondary School, Bromsgrove just east of Winston Churchill Blvd. (south of the QEW) on the third Thursday of the month, at 7:30 pm in the Little Theatre. For PET/CBM/VIC 20/Commodore 64.

TPUG makes every effort to ensure that meetings take place when and where scheduled. However, unforeseen problems may occasionally arise that lead to a particular meeting being changed or cancelled. The TPUG meetings line (445-9040) is the best source of fully up-to-date information on meeting times, and should be consulted.

Are you interested in organizing some other interest group in the Greater Toronto area? Please let the club office know, by mail, phone, or TPUG bulletin board.

OCTOBER

MON	TUES	WED	THURS
	1 VIC 20	2 Communications	3 Annual Meeting
7	8 Hardware	9 Central	10 Brampton
14 Eastside	15	16 SuperPET	17 Westside COMAL
21 New Users	22	23	24
28 Commodore 64	29 Business	30	31

NOVEMBER

MON	TUES	WED	THURS
			1
4 Eastside	5 VIC 20	6 Communications	7
11	12 Hardware	13 Central	14 Brampton
18 New Users	19 Business	20 SuperPET	21 Westside
25 Commodore 64	26	27	28 COMAL

Products Received

Presented by Astrid Kumas

The following products have been received by TPUG Magazine *in recent weeks. Please note that these descriptions are based on the manufacturers' own announcements, and are not the result of evaluation by* TPUG Magazine.

Look Sharp

Look Sharp from Mastertronic, distributed by Bullock Industries Ltd., 60 Bullock Drive, Unit 5, Markham ON, L3P 3P2. Price: $12.99 (Cdn.)

Look Sharp consists of two C-64 programs designed to sharpen children's observational skills and visual memory.

The first one, **Old MacDonald's Farm**, offers the choice of three different games: *Memory*, *Odd-one-Out* and *Snap*. In the *Memory* game, a picture is displayed showing animals on Old Mac's farm. The player has to put the same picture together from memory. The aim of the second game, *Odd-one-Out*, is to find out which picture of three presented on the screen does not belong to the set. The challenge of the last game, *Snap*, is to build up a picture of Old Mac by quickly pressing the right key when two pictures on the screen match. This game can be played by one or two players. In the one-player version, the computer becomes the opponent of the user. All three games offer two levels of diffulty, and are recommended for children 4-7 years old.

The second program on the disk is called **S.O.R.T.**, which stands for Space Observer Recruitment Test. It is designed for children 7 years old and up, and is described as also providing good entertainment for adults.

The program includes three testing games, all aimed at improving the player's visual perception. They provide an opportunity to practise one of the several visual skills (for example: matching pictures, spotting the odd-one-out, reconstructing from memory) required to succeed in the final **S.O.R.T.** test.

The Print Shop

Broderbund Software is releasing **The Print Shop Graphics Library, Disk Two**, the second follow-up product to its program, **The Print Shop** (reviewed in the May 1985 issue of *TPUG Magazine*.) **Disk Two** provides additional designs, symbols and pictures for do-it-yourself graphic creations, but this time in six new categories: Jobs, Hobbies, People, Places, Travel and Health. (A printer, disk drive and at least 48K of memory are required.) Suggested retail price is $24.95 (US).

For more information, contact Broderbund Software, 17 Paul Drive, San Rafael, CA 94903-2101, (415) 479-1170.

Still Sizzling...

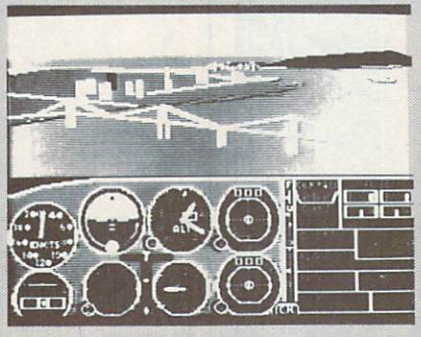
Stardos

Stardos, from Starpoint Software, is a powerful disk drive and Commodore 64 plug-in enhancement system. **Stardos** increases the speed of the disk drive 500 to 1000 per cent. The speed improvements apply not only to **LOAD**s, but to all disk commands and functions (including **SAVE**, **NEW**, **SCRATCH**, **VALIDATE**, **MEMORY-READ** and **MEMORY-WRITE**). The speed improvements do *not* require adjustment or modification of any mechanical part of the disk drive. **Stardos** may be used with any number of disk drives. It is completely compatible with any software library for the C-64, and also features many built-in utilities, such as a file-copy program that will allow the user to move program or data files to another disk; a disk copy program that will copy a whole disk in under three minutes; a mini word processor; a machine language monitor; and a disk editor.

The **Stardos** system is contained in a standard cartridge that you insert into the Commodore 64 expansion slot. The rest of **Stardos** is placed inside the disk drive itself. The user has to remove the disk drive cover, unplug the old ROM, and plug in the new one. There are no wires to cut, or extra cables to run to the computer.

For more information contact: Starpoint Software, Star Route, Gazelle, CA 96034, (916) 435-2371.

Scenery Disks



SubLOGIC Corporation announces the release of six different **Scenery Disks** for the Commodore 64. The disks expand the potential flying enviroment of SubLOGIC flight simulation products, including **Flight Simulator II** (reviewed by Dave Neale in the November 1984 issue of *TPUG Magazine*). **Scenery Disks** cover the entire western half of the continental United States. Each disk comes complete with appropriate sectional charts, plus full airport and nav-aid directories. Individual **Scenery Disk** packages are available for $19.95 (US) each, plus $2.00 for postage. The whole set may be purchased for $99.95 (US), plus $5.00 for postage.

For more information contact: SubLOGIC Corporation, 713 Edgebrook Drive, Champaign IL 61820. Order Line:(800) 637-4983.

Silent Service

Do you remember our review of **F-15 Strike Eagle** in the last issue of *TPUG Magazine*? Sid Meier (the author of the game) has created another simulation — but this time it's a submarine simulation. MicroProse Software's **Silent Service**, a simulation of World War II submarine combat in the Pacific, was scheduled for release in mid-September. Suggested retail price: $34.95 (US).

For more information contact: MicroProse Software, 120 Lakefront Drive, Hunt Valley, Maryland 21030, (301) 667-1151. □

I am the C-64

I am the C-64 from Creative Software, 960 Hamlin Court, Sunnyvale, CA 94089. Suggested retail price: $29.95 (US).

Creative Software has produced this series of two tutorial disks for those who prefer a hands-on learning method. Users who have their C-64s sitting at home, but don't have enough time and self-discipline to go through the Commodore 64 owner's manual, will find this product useful.

The **I am the C-64** program allows the user to flip back and forward through the pages. Starting with *Volume 3*, the authors provide the *Volume Index* to all the pages, enabling the user to call up to the screen any page he or she wishes to review.

The first disk, called the *Introductory Series*, contains three volumes: **Overall Introduction to the C-64** (Volume 1), **Introduction to the Keyboard** (Volume 2) and **Introduction to the BASIC Programming Language** (Volume 3). The approach is simple. First, basic concepts are explained; then one or two illustrative examples or programs are listed; and lastly, the user is asked to complete some instructions and run the programs as prompted on the screen.

The second disk presents the *Advanced Series* of **I am the C-64**, and also includes three volumes: **Advanced BASIC Programming Techniques** (Volume 4), **Sprite Graphics** (Volume 5) and **Music and Sound Effects** (Volume 6). All are intended for the user with some knowledge of BASIC.

Everything About The C-64

Everything You Can Do With Your Commodore 64 by Richard G. Peddicord, published by Alfred Publishing Co., Inc., 15335 Morrison St., PO Box 5964, Sherman Oaks, CA 91413. Price: $9.95 (US).

Everything You Can Do With Your Commodore 64, together with the subtitle *And How To Do It*, is a very accurate title for this book. It fits into the category of manuals or tutorials: it is very comprehensive and informative, as a good manual should be, yet — unlike most of the manuals on the market — it is fun to read.

The author introduces the C-64 assuming no previous practical or theoretical knowledge about computers on the part of the reader. The first chapters of the book describe how to hook up the system, then go on to explain the characteristics, capabilities and basic operations of the computer, as well as Commodore peripherals (cassette, diskette, monitor and printers). There is also a short introduction to BASIC programming. Users who are familiar with this material can skip the first five chapters and go to the ones that cover numerous applications (Games, Graphics and Music, Education, Word Processing, Financial Applications, etc.); and programming languages, hardware and other peripherals (Modems, Interfaces, Speech Synthesizers, Port Expanders, etc.). Together with explanations of different applications, the author includes recommendations on purchasing specific software programs and peripherals.

The book is well illustrated and well laid out: the essential information is highlighted, and there are many diagrams, as well as over 150 photos and illustrations of the actual products, which altogether makes the learning process much easier.

Handic B-128 Software

Handic Software has released B-128 versions of two popular software packages: **Calc Result**, a three-dimensional spreadsheet program, and **Word Result**.

For more information contact: Handic Software, 520 Fellowship Road, Suite B206, Mount Laurel, NJ 08054, (609) 866-1001. □

Bulletin Board

Keeping Victor Vibrant

Howard M. Mesick, the author of the article 'Keeping Victor Vibrant' in our August/September issue, has sent us updated information on sources of VIC 20 software.

Apparently, Stereo Discounters Electronic World, a company mentioned in the article, has sold out the last of their VIC 20 programs, and there is no guarantee that new stock will arrive. But another company, Kay-Bee Toy and Hobby (unfortunately, the name was misspelled in the article as K-B Toy and Hobby), is still well stocked.

If any VIC 20 users need a word processor that works with the 1526 printer, Howard tells us that **Quick Brown Fox** is still available for $19.95 US, plus $3.50 US postage and handling, from Micro-W Distributing, Inc., 1342B Route 23, Butler, NJ 07405. Cartridges without instructions, but with a keyboard overlay, are available for $9.00 US, plus a shipping and handling fee. Since the documentation runs to almost fifty pages and a cassette tape, he advises you to buy it, unless you already have a copy.

Commodore Technical Bulletin

Commodore has introduced a technical bulletin board, *TECHTOPICS*, announcing modifications, troubleshooting and other technical topics concerning Commodore computers and peripherals.

TECHTOPICS is available from the Customer Relations Department at Commodore, at a cost of $2.50 (Cdn.) per issue, including postage.

Among the topics that have been covered in the first seven issues are:

- troubleshooting tips for the 1702 monitor;
- specs and assembly upgrades for the 1541 disk drive;
- C-64 PCB assembly update;
- C-16 and Plus/4 troubleshooting aids;

Unclassified

This space is for the ads of TPUG members. Wanted or for sale items only. Cost is 25 cents per word. No dealer ads accepted.

Moving must sell: Commodore computer 32K, dual 4040 disk drive and manuals. Sound interface, 85 floppy disks loaded with programs in indexed binder inserts. Hundreds of books, manuals, subscriptions and associated literature. Best offer over $1,000 within one week. Ray (416) 267-8488.

For Sale: 4032 PET with VisiCalc and Tool Kit, 2031 disk drive, 8023P printer: Call (914) 783-7494.

TOTAL SuperPET PACKAGE: 3 board SP-9000, ideal for OS/99; CBM 8050 dual disk drive; 8023P Tractor printer; books and covers for all units; software includes Wordcraft 80 and several library disks; over $4500 invested. B/O over $3000. Don Siver, RD 5, Box 187, Coatesville, PA 19320. (215) 383-4664.

For Sale: SuperPET, 8050 drive, 4022 printer, 300 baud acoustic modem, cables, software and manuals. $2075, offers. Phone (519) 472-5621.

TPUG Magazine

Distributors

Dealers: *If you would like to carry* TPUG Magazine *in your store, you may order from any one of the following distributors:*

CANADA

Compulit Distributors, Port Coquitlam, BC 604-464-1221

USA

Prairie News, Chicago, IL	312-384-5350
Levity Distributors, North Hollywood, CA	818-506-7958
Whole Life Distributors, Englewood, CO	303-761-2435
M-6 Distribution, Houston, TX	713-778-3002
The Homing Pigeon, Elgin, TX	512-276-7962
Northeast News Distributors, Kingston, NY	914-382-2000
Fred Bay News Co., Portland, OR	503-228-0251
Alonso Book = Periodical, Alexandria, VA	703-765-1211
Cornucopia Distribution, Seattle, WA	206-323-6247
Guild News, Atlanta, GA	404-252-4166
Micro-PACE, Champaign, IL	800-362-9653

TPUG LIBRARY DISKS

10 months of the latest, fabulous public domain software

ONLY

$79.95* CAN

SAVE 20% or more by using this offer

Subscribe now to start receiving the ***TPUG disk of the month***

*US orders US$59.95

Name _____ Membership # _____

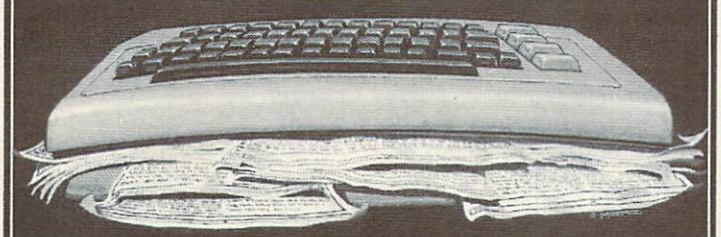
Address _____

☐ Cheque ☐ Money Order ☐ VISA ☐ MASTERCARD Amount $_____ Currency ☐ Can ☐ US

Credit Card # _____ Expiry date _____ Signature _____

Type of disk ☐ C-64 ☐ VIC 20 ☐ PET ☐ SuperPET (10 disk set)

Expand Past Maximum Capacity!



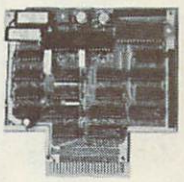
The Transactor

The Tech/News Journal For Commodore Computers

At better book stores everywhere! Or 6 issues delivered to your door for just $15.00 (Overseas $21 U.S. Air Mail $40 U.S.)
The Transactor. 500 Steeles Ave. Milton, Ontario. L9T 3P7.
416 878-8438
Also check out The Transactor Disk and The Complete Commodore Inner Space Anthology - to us, expansion knows no limits!

DIADACS 1 « » TAKE CONTROL WITH YOUR COMMODORE 64

16 CHANNEL 12 BIT
A/D CONVERTER
D/A CONVERTER 12 BIT
DIGITAL OUTPUT 10 LINE
DIGITAL INPUT 12 LINE
REAL TIME CLOCK



* 12 BIT RESOLUTION - A/D CONVERTER 30 Conversions per second. 0 to 10 volt inout. Controlled from Basic or Machine Language. * 12 BIT DIGITAL TO ANALOG CONVERTER - 0 to 10 volt output. * C-mos REAL TIME CLOCK CALENDAR - Year, day, month, day of month, day of week, hour, minute, second. Battery Backup - Set it once and forget it. * DIGITAL INPUTS 12 TTL compatible digital inputs for monitoring switches, relay closures, burglar alarms, etc. * DIGITAL OUTPUTS 10 TTL compatible outputs provide for controlling relays, digital displays, turning on alarms. On board +/-15 volts DC @ 20mA generated. Connectors for connection to external equipment. DIADACS 1 comes complete with user's manual and software driver.
Assembled and tested .....$279.00 US Dollars.
Add $3.00 for shipping U.S.A. Send check, money order, or MC/VISA.

MINI DIADACS 1 8 CHANNEL, 8 BITS A/D CONVERTER with 12 DIGITAL OUT and 12 DIGITAL IN.....$99.95 US DOLLARS

FAST DIADACS 1 SAME AS DIADACS 1 BUT HAS ON BOARD ROM SPACE and 100 MICROSEC CONVERSION.....$325.00 US DOLLARS

CONTROL MODE

New Software Package by TJN for the Commodore C64 and DIADACS 1. FEATURES:* DATA BASE MODIFICATIONS * ANALOG DEVICE DISPLAY *CONTACT INPUT DEVICE DISPLAY * CONTACT OUTPUT DEVICE DISPLAY * CONTACT OUTPUT CONTROL ** FUTURE SOFTWARE ENHANCEMENTS: * SEQUENTIAL CONTROL PACKAGE Will allow the user to build complex control sequences for home security, energy management and many other applications * ARCHIVE DATA STORAGE * REPORT GENERATION log of device values over several selectable time intervals using archival data. * TRENDING A plot of device value versus a selectable time interval.
CONTROL MODE I Without future enhancements.....$125.00 US DOLLARS
CONTROL MODE II With all enhancements.....$250.00 US DOLLARS

KMMM PASCAL

KMMM PASCAL for Commodore by WILSERV provides an EDITOR,EDITOR/COMPILER, and a TRANSLATOR. KMMM PASCAL is virtually a complete implimentation of Jensen and Wirth PASCAL and provides all of the string functions supplied by USCD PASCAL.
KMMM PASCAL on 1541 disk.....$99.95 US DOLLARS

CGRS MICROTECH

P.O. Box 102 • LANGHORNE, PA 19047
• (215) 757-0284

advertisers' index

TPUG Contacts

TPUG OFFICE 416/445-4524

TPUG BBS 416/429-6044

TPUG MEETINGS INFO 416/445-9040

Board of Directors

President	Chris Bennett	c/o416/445-4524
Vice-President	Gerry Gold	416/225-8760
Vice-President	Gord Campbell	416/492-9518
Past President	Michael Bonnycastle	416/654-2381
Treasurer	Carol Shevlin	c/o416/445-4524
Recording Sec.	John Shepherd	416/244-1487
	Rosemary Beasley	c/o416/445-4524
	David Bradley	c/o416/445-4524
	Doris Bradley	c/o416/445-4524
	Richard Bradley	c/o416/445-4524
	Gary Croft	416/727-8795
	Mike Donegan	416/639-0329
	John Easton	416/251-1511
	Carl Epstein	416/492-0222
	Keith Falkner	416/481-0678
	Louise Redgers	416/447-4811
General Manager	Louise Redgers	416/445-4524

TPUG Magazine

Publisher	Louise Redgers	416/445-4524
Editor	Nick Sullivan	416/445-9865
Assistant Editor	Marya Miller	416/445-9865
Production Manager	Astrid Kumas	416/445-9865
Ad Director	Louise Redgers	416/445-4524

Meeting Co-ordinators

Brampton Chapter	TPUG Office	416/445-4524
Central Chapter	Michael Bonnycastle	416/654-2381
C-64 Chapter	Louise Redgers	416/447-4811
COMAL Chapter	Donald Dalley	416/742-3790
	Victor Gough	416/677-8840
Communications	Darrell Grainger	c/o416/445-4524
Eastside Chapter	Judith Willans	c/o416/445-4524
Hardware Chapter		c/o416/445-4524
SuperPET Chapter		416/225-8760
VIC 20 Chapter	Anne Gudz	c/o416/445-4524
Westside Chapter	John Easton	416/251-1511
Business Chapter	Louise Redgers	416/447-4811
New Users Chapter	Louise Redgers	416/447-4811

Librarians

COMAL	Victor Gough	416/677-8840
PET	Mike Donegan	416/639-0329
SuperPET	Bill Dutfield	416/224-0642
VIC 20	Richard Best	c/o416/445-4524
BBS Sysop (voice)	Tom Shevlin	c/o416/445-4524
Assistant Sysop	Carol Shevlin	c/o416/445-4524

TPUG
LIBRARY DISKS

10 months of the latest, fabulous public domain software

ONLY

$79.95*CAN

SAVE 20% or more by using this offer

Subscribe now to start receiving the

TPUG disk of the month

*US orders US$59.95

Name _____ _____ Membership # _____

Address _____

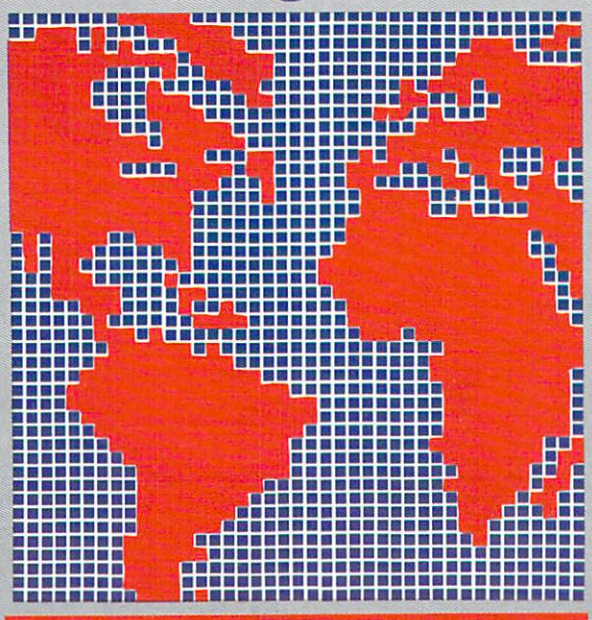
_____

☐ Cheque ☐ Money Order ☐ VISA ☐ MASTERCARD Amount $_____ Currency ☐ Can. ☐US

Credit Card # _____ Expiry date _____ Signature _____

Type of disk: ☐ C-64 ☐VIC 20 ☐ PET ☐ SuperPET (10 disk set)

THE WORLD OF COMMODORE III



The 1984 Canadian World of Commodore show was the largest and best attended show in Commodore International's history. Larger than any other Commodore show in the World and this year's show will be even larger.

World of Commodore III is designed specifically to appeal to the interests and needs of present and potential Commodore owners.

Everything about your present or future Commodore computer – from hardware to software, Business to Personal to Educational – from over 90 International Exhibitors. Price of admission includes free seminars, clinics, contests and free parking.

PLAN TO ATTEND

INTERNATIONAL CENTRE

DECEMBER 5 TO 8 '85

TORONTO

A HUNTER NICHOLS PRESENTATION

To exhibit contact:

Hunter Nichols Inc., 721 Progress Ave., Scarborough, Ont., M1H 2W7

(416) 439-4140