The Magazine For ALL Commodore Computer Users

TPUG Magazine

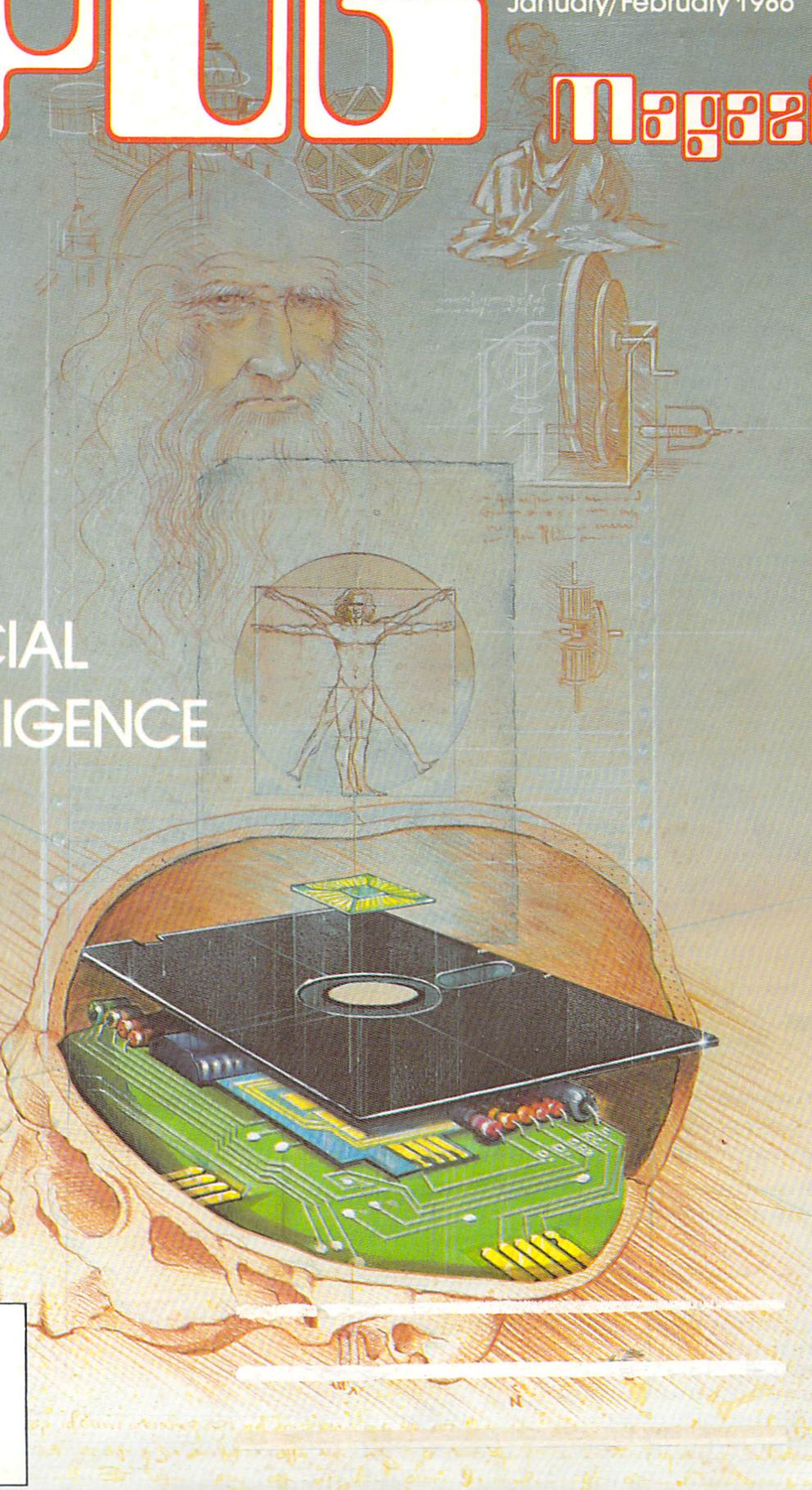
$2.95

January/February 1986

Douglas Hofstadter
A conversation on Artificial Intelligence

Punter BBS Commands
8-page pull out reference

Latest Amiga News



ARTIFICIAL INTELLIGENCE

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

UXB-Melbourne House Inc
C64
KUNG-FU
The way of the exploding fist.™
$29.95
NEW

subLOGIC Corporation
Jet
$54.50
Flight Simulator II
$58.00
Night Mission Pinball
$35.00

BATTERIES INCLUDED
PaperClip
The Word Processor.
C64
$76.95
With
SpellPack™
$92.95

EPYX COMPUTER SOFTWARE
C64

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

Brøderbund™

Printshop (disk)	$49.95
Karateka (disk)	$39.95
Printshop Graphics I	$29.95
Printshop Graphics II	$29.95
Printshop Graphics III	$29.95

Better Working
C64
B.W. Word Processor
B.W. Spreadsheet
B.W. File & Report
$59.95
Special Bundle All
3 For $140.00

INFOCOM

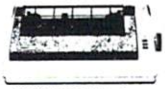
Enchanter	$39.95
Witness	$39.95
Infidel	$42.95
Planet Fall	$39.95
Sea Stalker	$39.95
Sorcerer	$42.95

C64
Newsroom $59.95
Clip Art $38.95
SPRINGBOARD™

Cardco
G Wiz Printer Interface for
the Commodore C64 . . . . $68.95
NEW S'MORE Memory Expansion . . . $69.95
(Adds 57% more memory)

Clearance Special
$9.95 ea.
4 for $35.00
Land and Sea Adventures
Adventure Master
Personal Spreadsheet
Personal Filekeeper
Board Game Challengers
(Real Estate,
Battleship, Solitaire)
Championship Gambler
Galactic Empire Builder
Star Trek Evolution
Chess Champion
Arcade Action
Ad Infinitum
Specttipede
Krypton
Olympic Skier
Westworld
Jungle Story
Bunny Zap
Subhunt
Zambeezi
Bionic Granny

Video 7
Master Modem
$79.95



SG-10 Printer
120 cps $429.00
Ribbon Replacements
6 pack $30.00
star micronics·inc

MicroProse Software
Acro Jet $38.50
Crusade in
Europe $44.00
Decision in
the Desert $44.00
F-15 Strike
Eagle $38.50
Kennedy
Approach $38.50
Silent
Service $38.50

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

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

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

If a product is not listed
call our Hot Line

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.
Prices subject to change without notice.

VISA MasterCard

Name: ____________________

Address: ____________________

City: ____________ Prov: ____________

Postal Code: ____________ Tel: ____________

Type of Computer: ____________________

soft-mail®

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

TPUG Magazine

Publisher: Bruce Hampson
Editor: Nick Sullivan
Assistant Editors: Marya Miller
Tim Grantham
Production Manager: Astrid Kumas
Editorial Assistant: Iwona Sukiennik
Cover Painting: Thom K. Wu
Creative Direction (Cover): Steve MacDowall
Cover Scan and Assembly: LK Graphics
Typesetting: Noesis, Toronto
Printed in Canada by: 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. All rights to material published in TPUG Magazine are reserved by TPUG Inc., and no material may be reprinted without written permission except where specifically stated.

Correspondence: Send change of address and subscription inquiries 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. Payment 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. Payment is made on publication. All contributions are subject to editing for length and readability. Address editorial contributions to: The Editors, TPUG Magazine, 101 Duncan Mill Road, Suite G7, Toronto ON, Canada M3B 1Z3.

Circulation
Subscription 12,000 Newsstand 10,000
ISSN #0825-0367

VIC 20, Commodore 64 and SuperPET are trademarks of Commodore Electronics Ltd. PET is a registered trademark of Commodore Business Machines, Inc. CBM is a registered trademark of Commodore Electronics Ltd.

Subscriptions to TPUG Magazine may be obtained by joining the Toronto PET Users' Group (TPUG) Inc.

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
Toronto 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 Sales Representatives:

Hunter Nichols Inc.
721 Progress Avenue
Scarborough, Ontario M1H 2W7
Telephone: (416)-439-4140

DIRECTORY

85 TPUG Magazine JAN

Feature: Artificial Intelligence

Articles

Reference Section

Micro Processes

Reviews

Departments

Inside Information

After two years before the masthead, and one year as Assistant Editor, Marya Miller has left our magazine to work as principal nabob of a weekly newspaper. As the real brains and chief inspiration behind *TPUG Magazine* during her tenure here, Miller was never one to let the allure of personal glory or worldly success distract her from the task of bringing you your all-but-monthly dose of Commodore-related information. Until now. We thank Marya for her extraordinary contribution to the magazine, and for putting up with us for so long; and we wish her every possible success in her new job.

Luckily for us, we were able, at very short notice, to blackmail regular contributor Tim Grantham into taking over the Assistant Editorship. You will probably remember Tim's feature article on C-64 music in our November issue (which, by the way, he will be updating next month), and his other articles over the past two years. Tim is also one of TPUG's sysops on the CompuServe Information Service, and one of the things he'll be doing for us is keeping track of the all the Amiga news and gossip that's rife on CIS these days. We welcome Tim to the magazine staff.

As part of this month's feature on Artificial Intelligence, we were fortunate in obtaining permission from the Canadian Broadcasting Commission to reprint the interview with Douglas Hofstadter, which appears on page six. Many thanks to Sara Wolch, producer of the CBC's program *Ideas*, for her unstinting cooperation. The complete text of the interview is available from the CBC, at a cost of five dollars (Cdn). The address is: CBC Enterprises, P.O. Box 500, Station 'A', Toronto, Ontario M5W 1E6. They also publish a free reading list on Artificial Intelligence — write to Ideas, P.O. Box 500, Station 'A', Toronto, Ontario M5W 1E6.

This month's cover is by Toronto artist Thom Wu, who also did the graphic accompanying the Hofstadter interview. The guy with the beard near the top of the cover picture is Jimmy-Bob Jackson, a charter TPUG member whose KoalaPad and **Doodle** pictures in the club library have made him famous in his own lifetime.

Last month in this space we foretold the appearance of detailed library documentation as an insert in this issue. It seems we spoke too soon. However, a library insert *is* coming, in the near future if not sooner, so don't abandon hope. This month we predict that all-purpose household robots will become an everyday reality for most North Americans by Christmas 1986.

Punter bulletin board systems are widely recognized as being the most powerful and popular single-user BBSs available for Commodore machines. If you want an idea just how much function Steve Punter managed to cram into his BBS program, consult the centre eight pages of this issue. The documentation there, written by Steve himself, covers *all* the BBS commands, with the exception of a couple of recent additions that we hope to cover in an update in a future edition. By the way, this month's reference insert is the second of what could be a long series, depending on response. Let us know what you think.

The editors □

A Message From TPUG

Please accept TPUG's sincerest apologies for any inconvenience you may have suffered in your dealings with the office in recent months.

Last August we moved to new offices to consolidate our operations and reduce our costs. In order to further consolidate, we reduced staff as well. This led to a backlog of orders and memberships to process in the fall.

In December, when we were just starting to get things back in order, our hard disk crashed, with our membership, invoices and Disk of the Month subscription series. Much of this data had to be reconstructed from hard copies.

These problems made us realize that the systems we have used at TPUG since its inception would no longer serve. It was time to switch to new, more efficient programs that would work faster and give us better access to the information we require. After some study, we decided to upgrade our computer hardware to IBM compatibles. Commodore Canada markets the PC10 — an IBM PC compatible — and we decided to go with this system. Implementing the new configuration brought its own share of troubles, however, and a delay of almost a month.

Acquiring the new hardware was only one step, however. Memberships were processed and subscriptions entered. Many hundreds of hours of work were done by volunteers at night and on weekends. Everything was finally caught up in mid-January. It was then a matter of duplicating the disks and mailing them out. By the time you read this, everyone should be up to date and have everything that they ordered.

As you already know, the magazine was also delayed by the move to new offices. We also had a chronic problem with advertising, which again was mostly the fault of an antiquated system. We have now hired a full-time advertising sales agency, and look forward to a much smoother operation. After the next issue, we should be back on schedule, and you should be receiving your issues before the cover date.

We appreciate the phone calls and letters we have received reminding us of missing magazines and orders. We have been doing our best to solve the problems but just have not had enough time to answer each one individually. Rest assured that we have looked at every letter and complaint and done our best to solve it as soon as possible.

We also appreciate the extra effort many of our staff and volunteers have given. Without this spirit of helping, we could not have got it done.

If anyone has not received their order yet, please contact the office as soon as possible, so that we may straighten out the problem. Thank you for your patience and your continuing confidence in us. We are here to serve you.

Bruce Hampson
TPUG General Manager □

ELECTRONICS 2001

AN AUTHORIZED COMMODORE SYSTEM DEALER SINCE 1979

POCKET MODEM

A 300 BAUD Telecommunication
Package for COMMODORE SYSTEMS

1200 BAUD
UPGRADABLE!

Package contains:

- one POCKET MODEM
- intelligent terminal software (C-64 version on disk)
- 6 ft phone cable
- documentation

Features:

- Auto dial / answer
- works with other terminal software
- C-128, C-64 compatible
- BBS included

C-64 & C-128 are registered trade marks
of COMMODORE BUSINESS MACHINES INC

$99.95

THIS MONTH'S SPECIALS

SIDEWAYS FOR C-64 **$39.95**	THE MANAGER C-64 **$19.95**	IMAGE MAKER MPS 802 HIGH RES. PRINTING **$10.00**	OUBLIETTE oubliette **$14.95**	SPRINT PRINT FULL SPOOLING PRINTER BUFFER SALE PRICE **$29.95**

The complete guide to the 1541 operating system.
$21.95

$36.95

PaperClip
(the word processor)

$69.95
with SpellPack
$89.95

Boss Joystick

$21.95

Kraft Joystick

$15.95

Compuserve Starter Kit
$39.95

PRICES IN EFFECT UNTIL MARCH 31, 1986

ELECTRONICS 2001 LTD.

Name ______________________
Address ______________________
____

□ Check or money order enclosed □ Visa □ MasterCard □ Amex

Acct.# __________ Exp. Date __________ Signature __________

ELECTRONICS 2001 LTD.

5529 Yonge Street, Willowdale, Ontario M2N 5S3
Tel: (416) 229-2700

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.

The Answer Desk

with Malcolm O'Brien

Changing disk names

I have been copying some disks that friends have lent me, using an Epyx **Fast Load** *cartridge on my Commodore 64. In most instances, I would like to change the names they have given their disks to names that I have chosen but, when I copy the disks, the names are automatically copied along with the data, overwriting the names with which my disks were formatted. My question is this: is there a way to change the name of a disk after it has been copied/saved? I must add that I acquired my 1541 only recently, and am not too familiar with it.*

Ron Charbonneau
Brampton, Ontario

There are a few ways in which you can accomplish this, Ron. Several utility programs in the TPUG library will enable you to change a disk's name (among other functions). Numerous disk utility programs are available commercially as well — in fact, you've already bought one of them! That's right, you can change the disk's name with **Fast Load**. Here's how:

First type the British pound sign, just as you do when copying, but this time select **E** (Edit Diskette). You will see on your screen:

TRACK 12 SECTOR 02

Remember that these numbers are hex, not decimal. This means that track 12 (hex) is actually a reference to track 18 (decimal), the directory track. Overtype the sector number to read 00, since we want to read the first sector of the track, where the disk name is stored. When you hit **RETURN**, the sector is read in, and the first 128 bytes are displayed on the monitor. The disk name starts at the 144th byte of the sector, so we need to look at the second 128 bytes. Just cursor down until the screen changes, then position the cursor on the byte labelled A090 (90 is the hex equivalent of 144).

On the right side of the screen you'll see the disk name. For example, if the name is **AAA**, the first three numbers on the line will be: 41 41 41. If you overtype these to: 42 42 42, you will see the name change to **BBB**. If you wish, you can take the opportunity to change the two-character disk ID at the same time (the bytes to change are 162 and 163 ($A2 and $A3), the two bytes immediately following the diskette name.

You have now done most of the necessary work. All that remains is to copy the changes you have made in memory back on the the disk. To do this, just type **W**.

Whizzer's Cattle

I am currently writing a game on my Commodore 64. I have the title screen and a few other screens done. Also, I have a problem: how can I move the bottom *of BASIC? I've seen this in one of my old Commodore magazines, but can no longer find the article.*

Chris Colohan
Oakville, Ontario

Using low memory for your graphics data is a good idea for two reasons. Reason one: BASIC will not overwrite your graphics with variables. Reason two: the VIC II chip can address only 16K at a time. All graphics data must be present in this 16K bank. There are four such banks of 16K in the 64, and the default is bank 0 — the first 16K of the machine. This is the area we're going to use after we move the BASIC program space to a higher memory location. This means that you won't have to do any bank switching, which will make things a little simpler.

To move the bottom of BASIC, you have to change the pointer in locations 43 and 44. Location 43 is the low byte and location 44 is the high byte. With this information we can find out where the start of BASIC is, and we can change it. Your BASIC programs usually start at 2049, and are preceded (at 2048) with a zero byte. The zero byte is necessary, so we'll need to include one of those in our reckoning too. Use this line:

```
poke 43,1:poke 44,64:pok
  e 16384,0:new
```

Now we have BASIC at 16384 ($4000). We have the required zero byte at the beginning of BASIC program space, and the **NEW** took care of adjusting other system pointers. The text screen is in its customary place between 1024 ($0400) and 2047 ($07FF). Other data (bitmaps, sprites and so on) can be located between 2048 ($0800) and 4095 ($0FFF), and between 8192 ($2000) and 16383 ($3FFF). Of course you also have the usual places for sprites in very low memory (704, 832, 896, and 960). You can *not* put graphic data between 4096 ($1000) and 8191 ($1FFF), because the VIC II chip sees character images in this area of memory, and your data will go unnoticed.

The next thing you have to deal with is loading your main program (to the new start of BASIC). The 'dynamic keyboard technique' is usually used to accomplish this. Here's an example:

```
1 if q$=chr$(34) goto 3
2 q$=chr$(34): print "<c
  lr><2 down>load";q$;"w
  hiz";q$;",8,1"
3 poke 43,1: poke 44,64:
  poke 16384,0
4 print "<4 down>run<hom
  e>";
5 poke 631,13: poke 632,
  13: poke 198,2: new
```

The main program can take care of loading the graphics data. It goes something like this:

```
1 a=a+1:on a goto2,3,4,5
2 load "bitmap",8,1
3 load "colour",8,1
4 load "sprites",8,1
5 print "Welcome to Whiz
  zer's Cattle!"
```

For this to work properly, you will need to have saved (as PRG files) the main program, the sprites, the bitmap and the colour, from the same memory locations that you want them to return to. Good luck!

Missing Cartridge Quest

I hope you can help me find a cartridge made by Thorn-EMI called Jumbo Jet. *I'm also interested in flight simulation cartridges or tapes. I already have* IFR *and the* Suzy Q *tape. I much prefer* IFR. *Are there others available for the VIC 20?*

Roger Martin
La Salle, Quebec

Software sources for the VIC 20 have all but disappeared. However, when tapes or cartridges can be found, they are usually priced very low. Recently I've seen bins of them at places like Eaton's, as well as the regular computer stores — unfortunately none of it was of the type you're seeking. The only other suggestion I can make is to watch the classified ads in your local newspapers or in *TPUG Magazine.* □

LINE NOISE

HAL timebomb

I have some important information for anyone using the **HAL BBS System** for the C-64. There is a set of secret commands written into Version 4.3h that allows any caller to wipe the message and user disk, and reset the computer. I have included the command sequence — please, do *not* print it.

I have revised the code to eliminate this problem and others, and will gladly send a copy of the revision (along with a copy of the uncompiled BASIC source code) to any registered owners of this package. Please send $3.00 (to cover the cost of the disk and postage), along with a photocopy of the front cover of your manual (showing the registration number).

Bob Swift
BOB's Border BBS
Hal #000098

Bob's address is: Suite 104, 530 Bannatyne Avenue, Estevan, Saskatchewan, CANADA S4A 2G5.

Two faces of Epyx

I am writing to inquire if I am the only one not getting any response from the Epyx Company on the defective **Fast Load** cartridges, and how to get through to the company.

Right after I received the June issue of *TPUG Magazine* and read the article 'Fast Load Lament', I wrote to Epyx explaining I must have one of their cartridges with the bug, and asking how to exchange it for a new one. The only reply I got was an ad from them to buy some of their software. In September, I wrote to Epyx again and all I got was another ad.

Howard I. Stearns
Alturas, California

It was most interesting to read 'An Epyx Saga' in the November 1985 *TPUG Magazine.* I have used two Version 3 **Fast Load** cartridges on two C-64 computers, one with a type 2 ROM, and the other with a type 3, since last spring.

All in all they have been most useful additions to the machines because the type of work I do with the computers requires frequent loading of different programs for various tasks.

There are, however, three bugs that seem to afflict the cartridges:

1) Any attempt to increase the number of relative files on a disk will cause errors, and may mess up the directory of the disk in question. This happened to me once, and my only recovery was to rewrite track 18, sector 0 using a 'disk doctor' program. All seems to be okay, though, if you first write to some large-numbered record without using **Fast Load**. Since this is often impossible with commercial programs, it is best to remove the cartridge before using those programs.

2) Part of the cartridge's speed seems to be because it omits certain built-in C-64 routines when loading. My experience with this is that screen garbage appears when attempting to load programs saved from a PET into a 64 with **Fast Load**. The program appears at 1024, thus filling the 64's screen with data! If this is a serious problem, just load the program once without **Fast Load**, then save it back to the 64. It will load properly after that (but it won't readily go back into the PET).

3) When using **Fast Load** as a DOS wedge, it is safer to include the drive number (for example, **@s0:filename** rather than **@s:filename**). I have a suspicion that there are times when the 1541 doesn't know that it's supposed to be drive 0, since I have had the occasional hangup when I haven't specified the drive number.

There are one or two other annoyances, such as the copy option's inability to transfer anything but program files, and the unusual monitor, but these are minor problems with a very useful product. I wouldn't be without it.

Don Colby
Brighton, Ontario

Several months ago, I wrote you about my V.1 **Fast Load** cartridge. It periodically scrambled my directory, erased programs, and generally created havoc.

After you published my letter, I also sent a complaint to Epyx, (simply addressed to EPYX, Sunnyvale, California — I had no further addresses). The post office must have had an off day — the letter actually arrived — and about six weeks later, I received free the V.3 cartridge. I never did send them the old one. I still have problems, albeit far fewer. Occasionally I get garbage when I call for the directory. Switching the drive off and on usually solves that problem, and the disk is not damaged. I still find every second attempt to format a disk produces garbage, and requires redoing. The new documentation is certainly far better for that portion of the cartridge utilities I actually bother to use. I must congratulate Epyx for their response, even though the cartridge arrived without a covering letter, or any explanation. The fact that they exchanged the cartridge is in itself highly commendable. For 99 per cent of my needs, the cartridge is excellent. For the few times it gets balky, I'm still way ahead.

I also need some advice. Is there a word processor program for the C-128 which is at least as sophisticated as **Paperclip** but also allows the creation and incorporation of graphs into the text? How completely compatible are IBM CP/M V3.3 program with my C-128?

Achim K. Krull
Agincourt, Ontario

The 1571 can read a variety of disks recorded with the MFM (Modified Frequency Modulation) technique employed by manufacturers such as IBM, Kaypro, and so on. If you have booted up the C-128 in CP/M mode, the 1571 should be able to read IBM CP/M software. However, you will not *be able to use programs intended to run under MP/M (CP/M for multiple machines) or CP/M 86 (CP/M for 16-bit machines).*

As far as we know, only the C-64 program **Newsroom** *from Scholastic Software can integrate graphics into text. However, its text editing capabilities are not nearly as comprehensive as those of a full-featured word processing program such as* **Paperclip**. □

TPUG Magazine *invites you to express your views on Commodore computing by writing to:*

Line Noise
TPUG Magazine
101 Duncan Mill Road,
Suite G7,
Don Mills, Canada M3B 1Z3

Exploring Brainy Gadgets

An interview about Artificial Intelligence with Douglas Hofstadter

*Douglas Hofstadter is an author, physicist and Artificial Intelligence theorist whose widely-published writings have made him the most renowned worker in the AI field today. His best-known work is the Pulitzer Prize-winning book **Gödel, Escher, Bach: An Eternal Golden Braid**, published in 1979 by Basic Books, which explores many topics relating to AI in a literate and highly imaginative style. His new book, **Metamagical Themas**, drawing on his writings in **Scientific American** magazine, is now available also. Last December, the Canadian Broadcasting Corporation's radio program **Ideas** broadcast an interview with Professor Hofstadter. What follows is a partial transcript, kindly made available to us by the producer of the program, Sara Wolch. The interviewer is **Ideas** host Lister Sinclair.*

* * *

LS: *Doug, how can I tell that you're intelligent, or conscious, or creative? How can I tell that a machine is intelligent, or conscious, or creative?*

DH: Those are good questions. On the other hand, I could turn them around and ask how do I know anything about you, whether you're conscious, whether you're creative, whether you're intelligent.

LS: *Also good questions, but I thought of it first (laugh). How, in fact, can we recognize these things in each other, or in any thing or person or animal or machine out in the real world?*

DH: Well, the way I look at it is that any of those things, consciousness or intelligence, but intelligence particularly, is a type of complex pattern, and we recognize it by looking for that pattern.

LS: *So I began by asking you about intelligence, and you seemed to change the subject because what you're coming back with is pattern recognition. Are you saying that it's really the same subject?*

DH: I am. In fact, I would say that creativity and consciousness and intelligence and pattern recognition are almost all synonymous terms. To me, the word 'pattern' and the word 'category' are very close. For example, a typical one would be the pattern or the concept of a vortex, a sort of a spirally thing that can happen in your bathtub when water is going out; or it can be part of a hurricane, where the hurricane is twisting; or it could be something in a magnetic field; or it could be the great red spot on Jupiter, or something. It might happen on a very small scale inside an atom. It might be just a vortex in a painting, or even a little twist on your head where you've combed your hair in a cowlick. And you could say all of those things are examples of a certain kind of abstract pattern that we call a vortex. But to me, you could also equally well say they are members of the category 'vortex'. So for me, the word 'pattern' and the word 'category' are really very, very close, if not exactly synonymous.

LS: *Do you have these things around your place? It's a blue card that I'm holding in my hand. It's about the size of an ordinary visiting card, and it has a number, and I'm warned on peril of my life not to lose the thing because it's the way we get into our building. Do you have those things?*

DH: Oh, yes.

LS: *And it's plastic, but inside is an electronic signal. And if you hold this up against the right box, the machine will open the door — or not, depending on whether it's the right card or not.*

DH: The problem with that kind of example is that it makes very clear distinctions between numbers that are allowed and numbers that are not allowed, or let's say patterns that are allowed and patterns that aren't. But there is a tremendous rigidity to the patterns that are allowed. The borderline of what is in and what is not in the category is far too rigid to be counted, in my mind, as a concept or a category. I will agree that it is in some sense a very rigid category, but it's not an interesting one. There's always a degree of open-endedness to any interesting category, as far as I'm concerned. But that's why I chose an example like vortex because we all recognize that there's some essence in there.

LS: *Some kind of twisty, spirally thing...*

DH: Right. But it's very hard to pin down exactly what that is, and it certainly isn't manifested in the same way in every different instance.

LS: *Okay. In other words, what you're saying is that even the way we define pattern is quite flexible.*

DH: Absolutely.

LS: *And there are many ways of defining different patterns, and often different ways of defining the same patterns.*

DH: When I gave the example of vortex, it did seem as if I was talking about the appearance of something. But I did not mean to imply that it's only the appearance. Obviously, we have many categories of things that are not even visual at all. I mean, all the different sounds that we would say sound French, for example. That's an example of a category — 'French-sounding things'. Or we could even go further and say 'French-sounding music', and we could say there's an example of a very abstract category, a very abstract kind of a pattern — the sound of French music.

LS: *By French-sounding things, you would mean particularly for an English speaker, the sounds that don't normally appear in English. For example, if we wanted to say 'a good white wine' in French —*

DH: "Un bon vin blanc.'

LS: *There we go, there go four nasals. All four sounds are foreign to English, and so 'un bon vin blanc' certainly sounds pretty French to me.*

DH: Yes, okay. You fooled me with that. I wasn't expecting to come out with the four nasals in a row. That's very nice, very nice.

LS: *Could I go back to the vortices for a* *moment? I think that's fascinating because it seems to me that what you're doing, Doug, is making analogies.*

DH: Agreed. Absolutely.

LS: *You're sort of making analogies in your head and almost treating the cowlick as being a kind of metaphor for the great red spot on Jupiter, and vice versa. So what you're saying, then, is that patterns don't even exist by themselves. Even the simplest pattern exists in a much wider context, and that must be a cultural context, or some kind of human context, I guess.*

DH: Yes, but the ability to abstract something out of its context, I would say, is something that is a universal ability of intelligence, and the reason that we have evolved and we have so far survived, survived successfully in this world, is presumably that our ability to extract patterns from the world is not a random one. We don't extract totally irrelevant things. We extract basically essential things.

LS: *Important patterns.*

DH: Important things from the world, ones that will serve us well in our day-to-day tasks. And if we were attending to trivial things all the time, we wouldn't have survived. So, I would say that it's all good to point out that patterns are in some sense subjective or depend upon the context, but at the same time, it's important to note that the pattern recognizers that have evolved and that survive are probably going to be attuned to similar patterns, because the criterion by which they're judged is whether that leads them to survive. And so I would assume that pattern recognizers of all sorts, even on different planets, would have found many of the same kinds of patterns as we have on this planet.

LS: *Could we look for a moment at what seems to have been a very primitive team, a very ancient team, namely the team of man and dog, in which each of the two creatures concerned is capable of a different range of complementary pattern recognitions. The man has good eyesight and the dog has poor eyesight. The dog has a great nose. So the dog is identifying scents, the man is identifying sights. One is much better at functioning in the daytime than the other. The two together, it seems, are a natural team, and sure, it's a natural team based on pattern recognition, based on categories. Each, so to speak, is dividing the world up into its own packet of categories. And yet the two together function as a very effective category apparatus, so to speak.*

DH: Yes. I don't think we have an analogous term to the word 'blind', as far as the sense of smell is concerned, but it is something like we are sort of — not quite, but almost — 'smell-blind'.

LS: *I like that, 'smell-blind'.*

DH: And it really is a completely alien experience to us. What must it be like to be able to have this rich input of all sorts of odours around? We can't imagine it. I could imagine having some kind of chemical analysis that is portrayed on a screen in front of us, and we try to use our eyes to play the role of a dog's nose. We're given a very complex graphic display, and we use a sense that we have, that is a very strong sense, to make sense of something that we can't do just with our own nose. It's sort of like the task that a computer has when it is faced with a wave form, when it's trying to understand or to — I guess 'understand' is the right word — to recognize the patterns of spoken speech. The computer doesn't have the same kind of built-in faculty as we do of recognizing speech, and it has to proceed through an extremely mechanical way where it is fed a waveform through a microphone, and it's as if this waveform were displayed on an oscilloscope, and then it has to try to figure out what is being said. You can imagine how hard that would be for us, if we saw a wiggly line on a screen, in front of us and we were told that is somebody speaking. What are they saying?

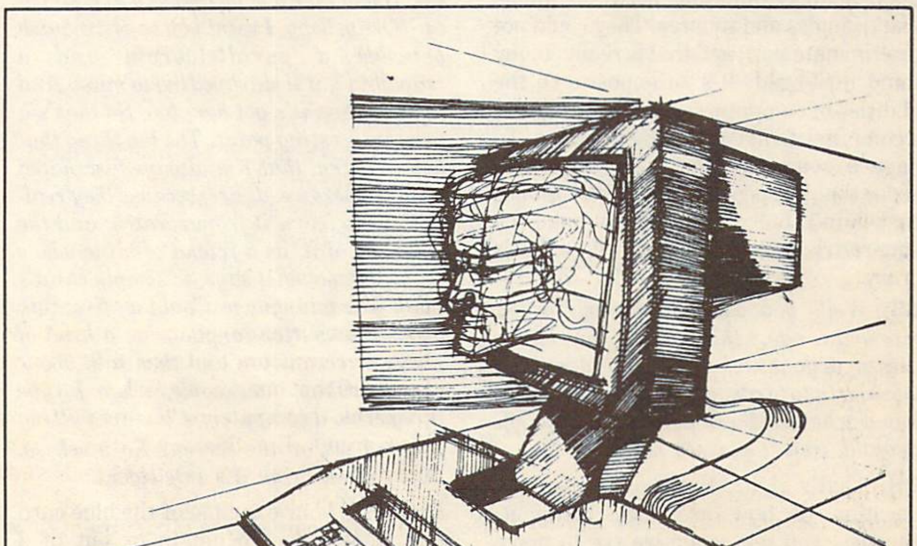
LS: *Oh boy, yes, that would be very rough, wouldn't it?*

DH: Yes. And that is the way computers have to deal with speech. It's similar to this idea of us having to deal with smells visually. To me, this idea of processing one kind of sensory input through another — I'm saying speech turned into a visual trace, and then looking at it visually — is a very interesting one.

LS: *I think you've done a kind of funny end-run here on us, because we suddenly seem to be talking about a kind of pattern which is surely a sign of intelligence, namely speech. And speech and language — that is pattern recognition of a very sophisticated kind.*

DH: Well, I have to admit that the patterns that fascinate me are the ones that are very abstract and very complex, and those patterns can really only be, almost by definition, recognized by intelligent beings.

LS: *There has been an experiment done with birds, who have very good eyesight, of course — eyesight is something they really rely on — and that's the 'canary experiment'. That's the one in which somebody took groups of canaries and trained them on chessmen. They had three or four pawns and a knight, and they put the food under the knight, which is a different shape. And pretty soon, the canaries caught on that that's where the food was. And then they made it several pawns and a bishop. It's now a different shape, all right, but not the same different shape. And it didn't take long, you know, very much quicker, the canaries learned that. And by a process of training, soon you could give the canaries four or five knights and a rook, say, and they would go for the rook. They rapidly understood that the food was under the different shape. And then, one day, they gave them*



half a dozen white pawns, all the same shape, and one black pawn — same shape again and a different colour. And sure enough, they went for the one with the different colour. Now that surely is forming an abstraction.

DH: It's a wonderful example. It's marvellous. I would have liked to know, however, what they would have done if there had been four white pawns, one black pawn and one white bishop — where you have one of a different colour and one of a different shape. That's an amazing example, though, the canary example. And another example of pattern recognition in birds, that when I first heard it, I found it almost impossible to believe — but I have heard that it's been reproduced, so I have to believe it; I really don't know what to make of it — it's the fact that a lot of slides were shown to birds, and birds would be trained that slides with a certain content would represent food, and slides with another kind of content would not. And the birds apparently were shown to be extremely good at discriminating between slides that showed pictures of people versus slides that had no people in them.

LS: *Pigeons, I think.*

DH: I think it was pigeons. You're right. And this meant that the people could be partially obscured, they could be inside a car, or they could be far away. There could be several people, there could be just one person. They could be very hard to make out, they could be wearing all sorts of clothes, they could be wearing bathing suits, whatever. But the amazing thing by contrast was that birds were not good at discriminating between things like triangles and squares. They could not discriminate that, and that is really, to my mind, just wild. It's so opposite to the abilities of computers. It's so hard to get a computer to make such distinctions. To have a computer that could recognize slides that had people in them versus ones that didn't, but was unable to make the square/triangle decision, is just — that's crazy.

LS: *Well, you said it yourself a few moments ago; that what the living creatures do is make important decisions. Obviously to birds, it's very important to know whether there are human beings around, and it plainly must be.*

DH: That's right. But it opens up the question of how are they doing it, because, you see, when we try to make computers do such a thing — make a very sophisticated discrimination between a slide with people — we consider that a sophisticated discrimination (people, non-people); and we consider the triangle/square discrimination to be a somewhat trivial one, because it's very geometric. And a geometrical shape such as a triangle is much more easily described formally or mechanistically than the distinction between person and non-person. So we in the field of Artificial Intelligence or Cognitive Science are trying to build up these primitive discrimination abilities. And so we think that anything that can do something sophisticated ought to be able to handle something trivial very easily. And yet it seems as if the birds are unable to do what we consider trivial, and are able to do something very sophisticated. I suppose human babies have that same quality. You'd be astonished if a baby could distinguish between square and triangle, but you wouldn't be very astonished if it could distinguish between faces or facial expressions.

The best chess-playing programs have an unbelievable amount of rigidity built into them...

LS: *It sure knows its mother from everybody else.*

DH: Exactly. So there's another example of, in some sense, the primitive abilities — or what we consider the primitive abilities — so deeply hidden inside the mechanism that they cannot be elicited at a conscious level.

LS: *If you take a baby and say to it sternly: "Okay, baby, I want you to distinguish between a parallelogram and a trapezoid", it'll start yelling to mum. And knows when it's got her, too. So that's a very interesting point. The big thing that babies learn, that I'm always fascinated by — (I guess we all are, because they really have to learn it from scratch, and the learning of it, as a friend of mine who's an anthropologist says, is simply child's play) — is a language. Could we first talk about the written language as a kind of pattern recognition and then talk about speech? Okay, how come, when I type things into a computer, it'll start putting things back on the screen? So much so, that I often think it's intelligent.*

DH: Well, your example of the blue card that allows you to get in or out of a building is very similar to the example of a computer being able to respond to certain key words. In fact the word 'key' is used: 'key word', 'key card' are very similar. You type a certain sequence — and now I'm going to have to say (I'm sorry): key *strokes*.

LS: *Is that what they're called?*

DH: Yes, 'keystrokes'. Yes, so you type a sequence of keystrokes and you create a word, and it's very easy for a computer to recognize a specific sequence of strokes. You know, 'A-B-C' or whatever it might be, some key word that it has been programmed to recognize. Because once again, it's a totally rigid distinction. It's really on-off — it's all or nothing. There's no blurriness.

LS: *Yes, if you misspell a word or something, as far as the computer is concerned, that's simply not the word.*

DH: That's right. Of course, there are programs that are to some extent capable of compensating. What I'm really saying is you can write programs — and there *are* programs — that will scan text and that will find words that are close to words that it knows, and will ask you whether you don't mean something else. In other words, they are capable of trying to compensate for spelling errors. Not nearly as fluidly or as flexibly as people are, because they don't know what's going on in the context.

LS: *But perhaps more attentively, in the sense they don't miss it, as a rule.*

DH: They don't miss it, but on the other hand, they miss things like the distinction between "its" and "it's", because both of them are correct, and so it doesn't know which one.

LS: *Ah, both of them are correct, but they're correct in different contexts.*

DH: And it doesn't look at the context, or even if it does look at the context, it may not be sophisticated enough to understand which word is really needed there.

LS: *When I was a kid, we often used to say "that seems pretty funny", and we would usually add "funny ha-ha", or "funny peculiar". Are you familiar with that one?*

DH: Oh yes, I did that too, as a kid.

LS: *Sure. Somehow we're aware that there's an ambiguity there, and the ambiguity has to be resolved, even for human beings. It must be really a nightmare for a machine, or rather for the human being who is devising the machine.*

DH: Well, the standard technique is to use certain kinds of rules of thumb, usually called 'heuristics' — rules which are not guaranteed to work, but which have a certain degree of probability of working.

LS: *Ah, like what our managers like to call 'guidelines'. Is that it?*

DH: Perfect. Absolutely. And the typical heuristic would be that in a context where certain kinds of things are being discussed, one meaning of a word is more likely to be meant than another meaning. It very often goes wrong. This is perhaps not the best example in the world, but if you have back troubles or something like that, and you're talking about something with the doctor, and you go too fast and the doctors says now, wait a minute, back up for a second. The word 'back' is there clearly not being used in reference to anybody's back.

LS: *He doesn't want you to get on your hands and knees and stick your back up.*

DH: Right, exactly. But the word 'back' has innumerable senses, and yet we're capable of making that distinction. I remember, in fact, the word 'sense' came up a little bit earlier when I was talking about the different senses that dogs and people have, smell and so forth, and I used the word 'sense', meaning 'meaning'. In the same sentence, I was using the word 'sense' with two different senses. The smell and vision and meaning, and yet, in that sentence, it was perfectly understandable. I mean I don't think anybody would have been confused. Except maybe a computer.

LS: *Ah. But no* body *would have been confused, but the computer conceivably could have been confused in that situation.*

DH: Yes, because the heuristics would probably have said, "Hey, look, we're in a context where we're discussing senses in the sense of smell, vision, et cetera, and so the word 'sense' should be interpreted that way."

LS: *Well, the written language and the problem, I suppose, of identifying the meaning of words, is one whole big chunk of pattern recognition, but speech is something even worse, surely, isn't it?*

DH: Spoken language is a fantastic problem because, as anyone who has ever tried to learn a foreign language knows, you can't tell where one word ends and the next word starts.

LS: *That's right. Foreigners don't know how to speak their own language. You read it in a book and you go to the country, and there they aren't doing it right!*

DH: That's right. It's amazing, I'll never forget when I went to Geneva as a 13-year-old, having studied French in junior high school, and I hear this phrase 'saillez, saillez', over and over again. And it's said so much as a unit, so clearly as a unit, that I was sure that it was an imperative of some verb. And I could not figure out what in the world people were saying, and I looked up the verb 'sailler' in the dictionary, and it wasn't there. I had no idea what people were saying. Only after a couple of months of being in Geneva did I finally somehow realize that what they were saying was 'ca y est'. It was three words, one after another, strung together, pronounced just totally fluidly, and it was a revelation to me.

A geometrical shape such as a triangle is much more easily described formally or mechanistically than the distinction between person and non-person . . .

LS: *Okay. Now, this is probably what's going on at home. We are talking to somebody, and there's probably only one or two people in the room, and there may be a lot of noise and other things going on, and a great number of signals, and this voice is coming out of the radio. And I have a suspicion that if we were to use, for example, the word 'sex', suddenly everybody in that room would pay attention, that that signal would cut through everything. And out of all those patterns, that one pattern has been identified. And that's one of the tricks in speech, isn't it? Can a machine do that? Humans do it all the time. Humans just did it . . .*

DH: Yes. Well, I know that in the early seventies, the Department of Defense in the US wanted to have programs that could do the monitoring of telephone conversations, and the idea was precisely that, of being able to recongnize certain key words as they occurred in telephone conversations. Words such as 'spy' or 'communist' or 'bomb' or 'terrorist', or things like that. And it proved to be a task that was way beyond computers at that point. They never reached the point where a computer could handle more than a few speakers using very restricted grammar and a fairly restrictive vocabulary under ideal conditions: that is, speaking directly into a microphone with no background noise.

LS: *Yes. But at this very moment, you know, there may be some kid sitting in a basement somewhere devising a machine to do that. I'm very wary of saying these things are impossible because I well remember people explaining to me, totally plausibly, I was totally convinced, that you would never get a machine to play chess. And I believed it because I couldn't play chess; and why, if I couldn't do it, certainly a machine couldn't; and suddenly now machines* are *playing chess, and playing chess very well indeed, I'm now told. And I am also now told that if the machine can do it, therefore it wasn't really intelligence after all.*

DH: Well, the slippery line does seem to be precisely that. One poses challenges to the computer scientists or the people devising programs to do things, and they wind up having a machine do specifically what you said it should do. And very often, even though it does, you feel somehow that that isn't really what you meant. I mean the ability to play chess — or you learn something about what you really meant . . .

LS: *Could you say that again? Because that's a very interesting remark. "You learn something about what you really meant."*

DH: Well, for example, this idea of getting a machine to play chess. If you find that a machine literally is able to play chess, that's one thing. And there are programs, as you said, that do play extremely well, but those programs are not modifiable to play other games. The best chess-playing programs have an unbelievable amount of rigidity built into them, and they can't move outside of the —

LS: *I'm sorry. Do you really mean rigidity built into them, or flexibility not built into them?*

DH: I guess you can put it that way, if you want. I think that, yes, nobody intended to build in rigidity, but in some sense it was just a by-product of the way they were designed, that there was an absence of flexibility.

LS: *Right. Whereas with my blue card that opens the door, they intended to build in rigidity, didn't they?*

DH: Oh yes, that's right. So what you find is that these machines that can play chess very well, they do literally what was prescribed, but when you consider the human ability to play chess, it also entails a sort of an aura or a halo around it: namely the ability to play chess-like games, the ability to do things that are *like* playing chess. And when you make a computer do it, you find, well, it can play chess and it can do nothing else, and this aura or halo of related things is missing. And that's a very, very big distinction between the way computers very often do things and the way people do them. □

A Piece of My Mind

by Avygdor Moise

Before I started to write this article, I searched in the dictionary for the definition of the word 'intelligence', and found the following descriptions: "mental brightness, being clever, quick in mind" and, yes, "department of state or armed services dealing with secret information".

Obviously, the above definitions relate to *human* intelligence. In view of the vagueness of the definition of 'intelligence', I suspect that many readers may have a false conception of the true meaning of the term 'artificial intelligence', as it relates to computers. It is therefore important to understand clearly what it is meant when we speak about artificial intelligence.

Natural intelligence

In order to survive, one has to learn to adapt to new and unexpected environmental and social conditions. Past experience is very valuable, since most of the decisions we make are based on similar experiences previously encountered.

When the adaptation process is biological in nature, and its effects are long-lasting (from generation to generation), it is known as 'evolution'. The evolutionary process will govern our ability to sense, analyse and respond to external stimuli. Our survival may therefore depend on our ability to react promptly to arbitrary events. If our reaction yields a favourable result, we memorize it for future use. If the result is unfavourable, the reaction is marked as a failure, not to be repeated should the situation arise again. This is the *learning* process. Our ability to learn, and to use what we have learned effectively, is referred to as 'intelligence'.

What makes the computer different, when compared to any other tools mankind has ever produced?

We think of tools as being extensions of our bodies. Tools amplify some of our capabilities, letting us accomplish tasks faster and more efficiently. Most of the tools we have built have been designed to accomplish a specific task, and cannot be converted to perform functions for which they were not designed: the tools are not *programmable*.

The computer, unlike any other tool, is a general-purpose instrument designed to perform tasks and control other machines, according to instructions that it was built to follow. The computer can be thought of as an extension of our brains, whereas all the conventional tools can be considered extensions of our limbs. The computer may thus control any man-made machine the way the brain controls our bodily functions.

The suggestion that the computer is an extension of our brain is really an ambitious one. People (and all other living entities) are complex creatures, and even the largest and most complex computer our current technology can produce is incapable of substituting for living tissue.

Computer intelligence

Even though our technology can't produce computers of a complexity and size comparable to the human brain, computer experts world-wide are researching software and hardware tools for the development of artificial intelligence and expert systems. As a result, a few computer languages were written to simplify program development.

Some of the languages used include assembler (pure machine code), LISP (and LISP derivatives), Forth, APL and other list-processing dialects. What makes LISP-like languages better programming tools than conventional languages (like BASIC and FORTRAN) in artificial intelligence programming? We'll answer this question with some programming examples.

Let's assume that we have chosen to program a general-purpose desktop support utility in Commodore BASIC 4.0. Starting with a two-function calculator, capable of addition and subtraction, the program might look like the one in Box 1. Here is a sample run (the computer's output is in bold type, your input in is regular type):

```
run
Ready? add
Enter x,y? 1,2
3
Ready? sub
```

Box 1 (An Unintelligent BASIC 4.0 Calculator)

```
1 gosub 10 :rem call application program
2 goto 1 :rem forever
10 input "Ready"; q$ :rem read function
20 if q$="add" then 90
30 if q$="sub" then 60
40 print "Syntax error, unknown ";q$
50 return
60 input "Enter x,y";x,y
70 print x-y
80 return
90 input "Enter x,y";x,y
100 print x+y
110 return
```

Box 2 (An Intelligent but Untutored BASIC 4.0 Calculator)

```
1 gosub 10 :rem call application program
2 goto 1 :rem forever
10 input "Ready"; q$ :rem read function
20 if q$="add" then 90
30 if q$="sub" then 60
40 print "Unknown ";q$ : gosub LEARN
50 return
60 input "Enter x,y";x,y
70 print x-y
80 return
90 input "Enter x,y";x,y
100 print x+y
110 return
```

```
Enter x,y? 1,2
-1
Ready? mul
Syntax error, unknown mul
Ready. . . .
```

This simulation performs as expected. The program is capable of adding two numbers and subtracting two numbers. When requested to multiply two numbers, however, it replies with the well-known message 'Syntax error' — it could not handle unexpected input *intelligently*. It is incapable of learning a new operation without being reprogrammed!

If PET BASIC allowed, the program could be slightly modified to handle unexpected inputs. The program in Box 2 is a thought experiment in Commodore BASIC. Here is how it might run:

```
run
Ready? add
Enter x,y? 1,2
3
Ready? sub
Enter x,y? 1,2
-1
Ready? mul
Unknown mul
Did you mean 'mul'? yes
Please define 'mul'?
(10) if q$="mul" then (40)
(20) print "Unknown ";q$ : gosub
LEARN
(30) return
(40) input "Enter x,y";x,y
(50) print x*y
(60) return () 'mul' defined
Ready? mul
Enter x,y? 2,3
6
Ready. . .
```

The program as we have now conceived it is self-modifying — if we list it after this run, we will find it mysteriously changed (see Box 3).

I do not intend to propose in this thought experiment a new method of programming in BASIC (though I believe that it is not too difficult to patch the command interpreter to actually execute the suggested program): I wish only to show the difference in the programming concept.

Normally, programs are composed of two distinct parts: the program section and the data section. The program section is the part that does not dynamically change: it cannot alter itself. If the programmer wishes to modify the construction of the program, he or she has to re-edit the source code. In contrast, the data section may be initialized at the beginning of the run, and the program may change the contents of the various variables.

In view of the above, it may be considered *good programming practice* to anticipate all potential inputs, then write the appropriate 'trap handlers' to prevent the program from accidental failure (illustrated in the first version of our BASIC program). In practice (especially if the program is very complex), you may not be able to foresee all the possibilities and write down all the actions that the program may have to take. In such cases, it may be beneficial to let the program *learn* as it goes along, enriching its capability through experience (some programmers may think of that as being self-debugging).

. . . The computer can be thought of as an extension of our brains, whereas all the conventional tools can be considered extensions of our limbs . . .

The second version illustrates this idea. When the program detects the unknown input 'mul', it asks the user to teach it to multiply. The moment the user types in the multiplication instructions, the program incorporates them into its code, where they are available for future use.

This ability to self-modify is what makes the program *intelligent*. The only way this can be made possible is to let the program read data, and then use the data to patch itself. In other words, the program executes the data.

The ability to bridge the gap between the program code and the data code makes it all possible. As you may be aware by now, programming languages like FORTRAN and BASIC will not permit the user to dynamically (at run time) enter a line of code and ask the computer to execute it as if the code was entered as part of the original program. LISP-like languages do not differentiate between data and program, making them suitable for artificial intelligence applications.

For example, the expression **(+ 1 2)** represents one list, which is made of the three elements (atoms) '+', '1' and '2'. If asked to evaluate this list, LISP will return the result '3'. Note that it will do it if (and *only* if) it is asked to evaluate the list. To further illustrate, the expression:

```
(eval (cons '+ (1 2)))
```

will construct the list **(+ 1 2)**, then evaluate.

To demonstrate some of the power of LISP programming, I chose TPUG's public domain version of XLISP (which runs on the SuperPET under OS-9) to write a simple calculator program (Box 4).

Here's an example session with this program (the computer's output is in bold type; the user's input is in regular type):

```
> (calc 'add 3)
  Teach me
    ( (n) ( (setq acc (+ acc n)) ))
  add
> (calc 'add 3)
  3
> (calc 'add 4)
  7
> (calc 'add 5)
  12
> (calc 'mul 6)
  Teach me. . .
```

Continued overleaf. . .

Box 3 (The Intelligent BASIC Calculator, One Run Later)

```
1 gosub 10 :rem call application program
2 goto 1 :rem forever
10 input "Ready"; q$ :rem read function
20 if q$="add" then 130
30 if q$="sub" then 100
40 if q$="mul" then 70
50 print "Unknown ";q$ : gosub LEARN
60 return
70 input "Enter x,y";x,y
80 print x*y
90 return
100 input "Enter x,y";x,y
110 print x-y
120 return
130 input "Enter x,y";x,y
140 print x+y
150 return *****
```

Note that the calculator program does not know how to calculate anything, but it is willing to learn. Once the algorithm for a given function is provided, it will incorporate it in its program.

Obviously, this program isn't perfect, but it was written just to illustrate the point. If you find the program and XLISP interesting, you may obtain a copy of XLISP from TPUG. XLISP was written in C, and will therefore run on your computer even if you don't have a SuperPET or OS-9 — as long as you have a good C compiler. □

Box 4 (A More Intelligent XLISP Calculator)

Note: the lines beginning with an asterisk are comments included for clarity — they are not part of the program.

```
 * Define a new class of objects called 'calculator' . . .
(setq calculator (Class 'new))
 * Define the calculator's internal variables . . .
(calculator 'ivars '(acc oplist keyboard fp hold))
 * Define the action to be taken when a calculator is first invoked . . .
(calculator 'answer 'isnew '()
  '( (setq acc 0)
     (setq hold nil)
     (setq keyboard "/term")
     (setq oplist '(learn isknown show))
     (print "Desk Top Calculator")
     self
  )
)
 * Define the response to the signal 'learn'.
   This is the INTELLIGENCE routine . . .
(calculator 'answer 'learn '(name / arglist function)
  '( (setq oplist (append oplist (list name)))
     (print "Teach me . . .")
     (setq fp (fopen keyboard "r"))
 * Input the method . . .
     (setq hold (read (fgets fp) ))
     (fclose fp)
 * Extract parameter . . .
     (setq arglist (head hold))
 * Extract function . . .
     (setq function (head (tail hold)))
 * Reprogram . . .
     (calculator 'answer name arglist function)
     name
  )
)
 * Find if the requested function is known . . .
(calculator 'answer 'isknown '(function)
  '( (member function oplist) )
)
 * Return the answer to: Does 'name' exist in 'names'?
(defun member (name names)
  ( cond
     ( (null names) nil )
     ( (eq name (head names)) t )
     ( t (member name (tail names)) )
  )
)
 * Create a new calculator and call it 'c' . . .
(setq c (calculator 'new))
(defun calc (function argument)
  ( cond ((c 'isknown function) (c function argument))
     (t (c 'learn function))
  )
)
```

Conference Date Change

The TPUG Board of Directors has decided that because of so many other conflicts in the spring (especially the nice weather), our annual TPUG Conference will be moved to the fall this year.

The new date should make it more convenient for many of our attendees. This year's conference promises to be even bigger and better than ever.

We are looking at many new, exciting and different ways of producing what has always been the highlight of TPUG's year. Look for further details in upcoming issues.

Turing and beyond

by Adam Herst

Since the turn of the century, psychologists have treated human intelligence as the ability to discern patterns, and to generalize and extend these patterns to novel situations. The extent to which someone possesses this ability is usually measured with the infamous Stanford-Binet and Weschler intelligence tests. These tests cover a broad variety of conceptual categories: verbal, mathematical and spatial, among others. The primary goal is the measurement of a general ability to manipulate symbolic information.

Intelligence tests must, however, impose a definition of intelligence before they can measure it. Such assumptions must be kept in mind when interpreting the results. Since we can't look into the human mind, we are restricted to studying the expressions of the mind through observable behaviour. The only way to study intelligence is through its expression, and that is usually limited to test-taking behaviour.

By this time, I've probably given you your fill of Introduction to Psychology, so let's turn back to computers. In the 1930s, a young mathematician named A.M. Turing was laying the theoretical foundations for the modern computer. As a result of his investigations into mathematical completeness and computability, he described what he called a Universal Machine, now known as a Turing Machine. This machine could solve any logical problem that could be characterized by a human problem-solver. In other words, if a problem could be solved using a limited set of procedures, manipulating a defined set of symbols in a finite number of serially-ordered steps, then a real-world implementation of a Turing machine should be able to solve it.

Your microcomputer is a real-world implementation of a Turing machine. Using a finite number of serially-ordered steps, the CPU of your computer manipulates a defined set of numerical symbols using a specified set of operations. A microcomputer is able to solve many of the problems that can be characterized by a human problem-solver.

In the early 1970s, it was realized that some of the functionings of the human mind could also be considered implementations of a Turing machine. Computer scientists, mathematicians and psychologists began to consider human intelligence — when defined as the ability to manipulate symbolic information — as the functioning of a Turing machine. It is this parallel that has encouraged the attempts at reproducing human intelligence on a computer. If the set of procedures and symbols used by the mind can be explicitly stated, then any Turing machine should be able to replicate the results they achieve.

Given the logical possibility of reproducing human intelligence on a computer, how would artificial intelligence be recognized? If we identify human intelligence through observed behaviour, then the same behaviour should identify computer intelligence. If a computer's behaviour measures up favourably against that of a human, then it is fair to say that the computer exhibits intelligence.

Imagine that you are in a space ship stuck in an earth orbit. The ship is equipped with the most sophisticated computing devices, capable of English language information exchange, and sensitive to the emotional side of human beings. All of your communications are cut off, both to the computer and to ground stations on earth.

Suddenly your computer terminal activates, and you are able to establish communications with someone named Hal. You begin an extended conversation, during which Hal enquires about your physical and emotional well-being, and expresses concern over your situation. Eventually, you find out that a method has been found to return you to earth. Before you can learn more, the communication stops, and you are returned to your solitude and left to wonder as to the source of the transmission. Was it the folks back home, or your onboard computers? From the information you received on your terminal, you would not have been able to distinguish whether you were talking to a man or a machine. With the proper degree of sophistication, computers and their programs can simulate seemingly intelligent conversations with humans.

Attempts have been made to program computers according to the principles of accepted theories of human intelligence. In the field of Cognitive Psychology (the study of learning, memory, thinking and reasoning), programs have been developed that simulate accepted theories of recognition, the learning of conceptual definitions, and the processes of deductive reasoning. These programs strive to achieve human-like behaviour from computers, using models of human cognitive processes.

More successful attempts at simulating human intelligence have tended to come from theories expressed through mathematical algorithms dealing with subsets of intelligent behaviour. Their success may be due in part to the fact that such theories are expressed in the natural language of the computer, numerical symbols. As pure Turing machines, computers are ideally suited for the serial manipulations required by mathematical procedures.

While the human mind may also deal in numerical symbols, a strong case can be made that it also deals in other, qualitatively different types of symbols. Numerical symbols do not seem suitable for the manipulation and expression of emotional concepts, for example. The human mind is not restricted to serially-ordered manipulations. It is capable of processing many types of information at one time, and of processing a single piece of information in many different ways. It would seem impossible for computers as implementations of Turing machines to reproduce all aspects of human intelligence.

The only constant in the computer world is change. If you have owned a computer for more than two weeks, you have probably noticed that you can now buy the same machine for half the price. You can expect attempts to be made to overcome the limitations of serially-processed numerical symbols in artificial intelligence. Some researchers look to parallel processing as a means of simulating the workings of the brain more closely. For the same reason, another trend has been towards declarative rather than procedural programming languages. At the heart of these attempts is a conviction that electrical impulses in an inanimate system can simulate electrochemical impulses in an organic system. □

Artificial (Fake?) Intelligence

by Jim Butterfield

Artificial intelligence (AI) is a busy buzzword this season. And I fear that the concept behind the term is hidden within the myth generated by movies, books and the press. It makes good fiction for a computer to sulk, or boast, or worry about whether it's getting old. And in the film *2001*, it's moving when HAL 9000 says: "My mind is going, Dave... I can feel it."

But that's fiction. And, while artificial intelligence is a real branch of computer science, it has little to do with making computers more human-like ('anthropomorphic'). The real computers that we know and love (or curse) are mechanical drudges that just follow instructions placed in their memories. Cleverly written instructions can cause the computer to do seemingly clever things, like beating you at chess. But you always know that it's just a dumb machine.

We often assign personalities to machines we use frequently. Betsy the car may be getting a little old and shaky, but she's a game old girl just the same... When we say such things, we don't really believe that an automobile has human characteristics. Similarly, we may identify computers as being friendly, comfortable or serious, but we don't really think of them as living things.

There are a number of clever programs that imitate human behaviour. Perhaps one of the earliest of these was **Eliza**. This program imitates the role of a psychologist, mostly by echoing things you say. For example, you could type in *I am gloomy* and the program might respond *Why are you gloomy?* If you continued from there with *I can't get my act together*, the computer might say *What would you do if you could get your act together?* But it becomes easy to see that the computer has no intelligence in the usual sense — typing *I am etaoinshrdlu* will get a reply such as *Why are you etaoinshrdlu?* It's not really paying attention.

Eliza is polite: in contrast, a similar program from Great Britain performs a similar task in a more aggressive way. Perhaps it's a result of the National Health Service, but **Insulter** starts by asking *What do you want, you miserable worm?* and responds to your input in a similar vein. *I am gloomy* might be answered with *How dull, so you're gloomy*.

A quite different style of 'intelligence simulation' may be found in Activision's **Little Computer People Discovery Kit**. The purpose of this program is to set up communications between you and the little person who lives inside your computer. Once you make contact, you may chat with the little person to find out his or her name and other details, such as height, age or clothing. The little person would usually rather play games, and will tap on the glass of the screen to get your attention. Strangely enough, each computer houses a different little person; it's odd to hear members of a user group discussing what their various little people are like.

None of these programs shows intelligence on the part of the computer. Despite their charm and simulated personality, the programs are doing what their programmers have planned. The inclusion of a 'randomizing' feature makes such a program more variable, but it cannot do anything new.

There is such a thing as artificial intelligence, however, and it's easy to write simple programs that have it. Here's a very simple definition: an AI program is one that learns as it runs. That way, it can draw its information from the user, not from the programmer. The user often doesn't know that information is being supplied, but it is.

Take the simple program **Animals**, for example. In this program, you are asked to think of an animal, and the computer will guess it. At the start, the computer knows only two animals, say a horse and a fish. When you signal that you've thought of an animal, the computer might ask *Does it live in the water?* If you reply *No*, it will guess *Horse*, and if you say *Yes*, it will guess *Fish*. It then asks *Did I get it?* And if not, it will ask what the animal was. Suppose you had thought of an elephant (answering *No* to the first question). The program would then ask *How can I tell a horse from an elephant?* you would supply a distinguishing question, such as *Does it have tusks?* Then you

How can I tell a horse from an elephant?

would explain that, for the elephant, the answer would be *Yes*.

The computer now knows about an elephant, something that the programmer didn't tell it. As more questions are asked, new animals will be added to the knowledge base: birds, insects, dinosaurs. Leave the program running in a classroom for a day, and the computer will know all the animals that the students know... plus a few extra ones that were just invented or misspelled. Leave it at a computer show for a couple of days (with a suitably-sized disk to catch all the data) and you'll be amazed by the menagerie that it collects: everything from hobbits to Easter bunnies.

Here's the strange thing: the computer ends up knowing more animals than any single person... certainly more than the programmer knew. It has assembled all that knowledge into a massive database. That's computer intelligence.

Again: if the program was given a large number of 'possible strategies' for winning at the game of checkers, that would not make it a very good player. But let that computer play lots of people — and tell the computer that every time it loses a game, it should mark the strategies it has used as 'lower priority', and every time it wins a game, it should mark the strategies it has used as 'higher priority'. After a while, the computer will play a pretty good game of checkers. It has learned, not from the programmer, but from the people it has played.

That's the essence of AI. A computer that can modify its behaviour based upon its experience is a program that gets smarter as it is used.

Perhaps the big payoff for artificial intelligence — at least one of the earliest — is that of 'expert systems'. Get a group of experts on a given subject — say, medicine — to input their knowledge in a certain area. Have a computer program that's written well enough to link together the elements of information in an intelligent way (including the contradictions, if any). Now you can have a computer information base — an expert system — that knows more than any one person. It can change, expand and be corrected as necessary, and will continue to gather knowledge.

In a sense, I view the essence of artificial intelligence as the concept that a program can become more knowledgeable than its programmer. Which brings to mind a new way to view the old proverb: "Computers are dumber than people... but smarter than programmers". □

Expert System, Novice User

by Dave Powell

I'm not an expert on expert systems. I suppose I know as much as many data processing professionals or computer hobbyists who read a variety of articles. My conception has always been of a system to which an expert could impart knowledge so that a layman could avail himself of it without special training. The articles often cited medical knowledge as an example. I imagined a doctor imparting his or her expert knowledge by typing in 'patients who sneeze have a cold or an allergy'; or 'normal temperature is 98.6 degrees Fahrenheit'; or 'red spots are a symptom of measles'.

I never really visualized the input process, but I suppose I thought that it had to be pretty simple, or maybe done by a layman who was, however, expert in using the expert system. Somehow or other, the system sorted out all the input and worked out that 'red spots' were a skin condition, and that 'measles' was a disease. After the (medical) expert introduced a new word or phrase, I imagined the system checking back: "'Measles' is a disease, right?" (on the grounds that it had 'symptom of' in front of it); "How are these red spots different from acne?"; and so on. Statement, question and answer, until the expert decided that the system was sufficiently educated.

I also expected that the expert system would keep growing, unless it was an inherently restricted set of knowledge, such as a well-explored branch of mathematics. It would require an easy way to add knowledge on a continuing basis.

Along came **XPER** from Abacus Software — an expert system that runs, not on some million-dollar mainframe, but on the Commodore 64. I thought I'd teach it how to diagnose ailing garden plants. Early on, I found out who had to categorize and input the information — me. I described several features by listing them, using the supplied editor:

- Insects visible
- Leaves have holes
- Leaves have altered colour
- Type of plant
- Time of year

That's enough to give you the idea. Each feature can have a maximum of fourteen attributes. For example, the first one, Insects, had:

- Colonies of small insects
- Small pink insects, masses of white 'cotton'
- Green beetles

And so on. 'Leaves have holes' has attributes that describe different shapes of hole. (Time to own up: all of this expert knowledge came from the 'Plant Disorders' chapter in *The Reader's Digest Illustrated Guide to Gardening*, which is very nicely set up to convert to the form **XPER** requires.)

Having set up features and attributes, the objects came next. I chose to use the name of the ailment as the object. Now comes the expert bit — connecting the objects with the descriptions. Object by object, I had to tell **XPER** which attributes each feature has. For instance, I added 'aphids' as an object. For the 'insects' feature, I chose *Colonies of small Insects*. I chose none of the attributes for the holes or altered colour features, 'all' for *Type of plant*, and 'spring' and 'early summer' for *Time of year*. Note that it's possible to choose none or several of the attributes for a given feature.

This part gets a bit tedious. In fact, I cut down on the number of objects and features that I was going to test, in order to get it done. My expectations of the expert system were changing!

Now to use it. Before leaving the editor, it's necessary to save the knowledge base, as it's called. Then load the 'inquiry' part via the main menu, and reload the base.

Okay, let's say I've just come in from the garden and my roses are looking a bit sorry for themselves. I want to find out what ails them. I can start with any feature and choose which attribute(s) describe what I've seen in the garden. A good place to start is the *Type of plant*. I'll choose both 'roses' and 'all'. **XPER** tells me immediately how many objects (ailments) I've eliminated, and how many remain. I can list either set. For the eliminated ones, I can ask why. Not much use after one question — anything eliminated was eliminated because it attacks plants other than 'roses' or 'all'. If I list Colorado Potato Beetle, I'll see that the little beastie prefers eggplants, peppers, tomatoes and white tomatoes.

On to another question. Let's say it's early summer (don't I wish!): I can answer *Time of year*. Possibly this question doesn't distinguish between any remaining objects. In one mode — the default — this question (feature) would not even be listed (nice touch, that). Continuing, I choose *Insects visible*, and answer *Colonies of white insects*. Aha! says **XPER**. Aphids. (Actually, it says 'object found').

Suppose I suspected aphids in the first place (even I can recognize aphids). I could ask **XPER** to optimize the search — for aphids. It will arrange the features in the order that will select aphids (if that's what I've got) in the least number of questions. It will go straight to *Insects visible*, and if I answer *Colonies of white insects*, it will tell me 'aphids', because only aphids have the attribute *Colonies of white insects*. Got the little beggars in one. But I'd hate my doctor to diagnose me with a single question.

When I tested my knowledge base, it picked out the ailment my wife described each time: but then, she was describing the ailment from the same book that I'd used to build the base. Like me, she thought she'd rather have just used the book in the first place. The book has pictures — which is how it gets away with the 'colonies of small insects' description.

Some observations

First, to build a base that would cover all the ailments that I might expect to see in my garden would take for ever, and might be beyond the capacity of the system. Secondly, it became obvious that my first design was not too good. There are too many possible plant types to fit in fourteen attributes, and there should be an 'umbrella' category ('symptoms', maybe) above the first three features. In an expanded knowledge base, there would be far more symptoms.

There's a way to fix these two problems, but there seems to be a bug which stops me. A father-son-connection feature looks as if it should work like this: initially, the feature 'plant type' would contain attributes that were subsets: vegetables, shrubs, trees, and so forth. Picking 'shrub' would open up a new feature, *Shrubs*, which in turn would have individual shrub types (juniper, yew, et cetera) as attributes. Irritatingly, even the example in the manual led to a system reset every time I tried to define this.

Depending on what changes are necessary, a redesign could mean a great deal of work, so continual testing is necessary, and probably some initial thought before even sitting down at the C-64.

How good is the product?

An early thought that entered my head — this was merely an in-memory binary database. Imagine a record for each object. Each feature is a field, and each attribute is a yes/no (1/0) byte, or even bit in the field. Now, to select an object, select all records that have a 1 in the right place. Then on to the next question, and do the selection only on the records already selected. Having every record in memory makes it fast.

This is a viable model of what is actually happening, but forget any ideas of emulating the process with database software. The process would take too long with a normal database, and making changes would be prohibitively difficult. **XPER** allows the base designer to add or remove objects, features, attributes and connections at any time (in the editor).

From a technical point of view, **XPER** is not so hot. I dislike the fact that switching from editor to inquirer takes so long. It also needs a disk swap, which is left up to the user. If one leaves the data disk in when the system disk should be there, there are circumstances where the program just does a cold restart. There doesn't seem to be much in the way of disk error checking. A save could fail — write-protect on, say — and no message would be given. In other cases, when an error message *is* given, it flashes on for one and a half seconds only. There is no facility for getting a directory of knowledge bases from the data disk — one has to get a directory before loading **XPER**.

There is an unusual mixture of menus and commands. Help 'screens' are a cryptic one or two line list of available commands. Okay for experts, but it sends beginners back to the manual. Strangely enough, it seems to work well once a familiarity is obtained. The function key arrangement works well, too.

Other C-64 features are not so well-used. The standard 'use the F keys to change the colours' option is available at the main menu, (with the keys mislabelled!), but then colour is avoided. Good use could have been made of colour in screens where the list of attributes is shown. The attribute selected by the user is in reverse, the one for the object just eliminated is shown with a 'dollar' sign in front of it. Why not use a different colour? There's no sound at all — better than programs that beep at every key, but a well-placed beep is a useful way to alert the user to mistakes. It also ignores capitals (even though they are used in the manual's examples), and some punctuation marks.

Lastly, **XPER** is not crash-proof. Many user errors or product bugs lead to a system reset. This can be particularly upsetting when the last hour's work has not been saved.

The manual is fine, as far as it goes, but gives no hint of why one would want to use the package. There is a pretty good example to follow, but there are occasional page-long passages of keystrokes that have to be entered correctly, or the example following will fail. (It failed for me a couple of times before I got it right.) There is a reference section, arranged by screen. Each screen has a four-character ID, which is not used by the system, as far as I can see, so why not spell out the function in full? The manual does have some good cross reference sections in the appendices, although some page references are wrong.

Conclusion

As a sophisticated software product, **XPER** doesn't do anything for me. As an expert system generator, it's really interesting — great for experimenting with abstract concepts such as the nature of information. It allows information about a fairly small body of knowledge to be stored, dissected and used. Pioneering a new technology is never easy, and **XPER** is a brave attempt. I'd like to see some technical improvements and, if possible, a larger capacity. Hopefully, later versions won't have the bug I encountered or, if it was my fault, will have a better explanation of how to do what I want.

For the person who wants to experiment with the concept of expert systems, the flaws will seem trivial and the potential great. As for my gardening problems, I'll stick to the book! (It occurs to me that an expert system becomes an improvement over text books at the point where more than one text needs to be referenced — it's a good way to consolidate the information.)

Finally, I suspect that the future of products like **XPER** will hinge upon the availability of commercially-compiled knowledge bases. Then I might be able to buy a diskette with a *Gardening Problems* base ready-made!

XPER Knowledge Base Management System for Microcomputers, from Abacus Software, PO Box 7211, Grand Rapids, Michigan 45910. $59.95 (US). □

Amiga Dispatches

by Tim Grantham

The best thing about the amazing Amiga is not its multitasking or its digital sound synthesis or even its hi-res colour graphics. The best thing about the Amiga is that it has brought some *excitement* back into the personal computer scene. Consumers pester their dealers for the latest software releases. Dealers hotly debate the cost effectiveness of the Atari ST versus the Amiga. Modems are smoking as software developers stay up to the wee hours of the morning trading tips on the latest versions of C compilers. Programmers haven't had this kind of gleam in their eyes since Jack Tramiel sprang the Commodore 64 on an industry that has since grown moribund with the suffocating dominance of IBM. It is great to see that Commodore has stuck to its tradition of innovation. Whether the Amiga gets the sales it deserves remains to be seen. Meanwhile, the latest news from the Front:

Copy Protection

For a time, the hottest topic of debate on Compuserve's Amigaforum was the issue of copy protection (CP) on Amiga commercial software. Electronic Arts, the independent software developer with probably the greatest commitment to the Amiga, invited members to get their feelings about CP out in the open. They responded enthusiastically. The following is just one example (edited):

#: 2093 S1/Software
17-Dec-85 11:45:37
Sb: #Open Systems & Software
Fm: William Volk(Aegis Devel 72257,2566
To: ALL

. . . but needless to say, we feel that copy protection has no place on a professional multi-tasking computer. This doesn't just apply to productivity and utility software . . . why shouldn't a user be allowed to play an adventure game while their spreadsheet is printing? (or their program is compiling). If the Amiga doesn't have open software . . . the whole point of the multitasking will be lost and the computer will have lost a real feature over other 68000 based systems. I want to stir up a little debate here . . . Let's hear it for "OPEN SYSTEMS AND OPEN SOFTWARE!!!"

Copy protection on Amiga software means, of course, it is usually impossible to install the program onto a hard disk. For many, an Amiga without a hard disk is like a juggler on tranquillizers — nothing's dropped but nothing flies! Electronic Arts also installs a custom OS on some of their games especially as a means of copy protection; but this also effectively short-circuits AmigaDOS's multitasking. Electronic Arts responded with:

#: 2155 S1/Software
17-Dec-85 22:29:00
Sb: #2132-#Electronic Arts & CP
Fm: Electronic Arts 76004,237
To: GARY SARFF 70167,2216 (X)

Actually, only a few of our programs will 'take over the systems', and they are all games. We try as hard as possible to play fair and share and, once we understand the system better, we'll probably do a better job. Unfortunately, many of our initial releases had to be developed while only the more primitive features of the OS were ready and bug-free, so they were not at as high a level of system interface as they could have been . . . We do realize the advantages of the multitasking, but Speed Demons who need the performance don't want to have an OS taking its time to decide who gets to use the system. Again, really just the games which don't create files, use outside data, etc.

JJ — EA

Some-one in a back room at Electronic Arts suggested one unorthodox means of CP: a registered purchaser could return their copy to EA, where it would be deprotected and their credit card number written into the start-up screen. That software would never get out of the safe-deposit box! After some more thrust-and-parry, EA's last word was:

#: 4811 31/Software
10-Jan-86 18:14:04
Sb: #4741-Copy Protection
Fm: Electronic Arts 76004,237
To: George Gregori 72256,3204 (X)

We always have and probably always will protect our games. I have yet to hear a valid argument against that. Our utility programs, especially the Deluxe series, are being released protected for now. However, we are doing lots of research into alternative methods, and are packaging our next release of DPaint with two copies already, plus the ability to purchase more backups. Some of our alternative methods include: starting the program by displaying a black & white digitized picture of the package cover, and having the user tell the program what color a particular object on the package is; (2) Having a special phone-order for non-protected disks and having that user's copy programmed to display his Visa/MC card # upon bootup; (3) Having two separate versions, mass market (cheap and protected) and specialty (expensive and non-protected).

Any other suggestions for alternative protection are greatly appreciated. Please don't ask us to trust all of you completely. We've been burnt too often by pirates in the past. But we're trying.

JJ — EA

Hardware News

On the hardware front, considerable interest was stirred with the revelation that v1.1 of AmigaDOS is compatible with other processors in the 68xxx series. It seems you can switch the 68000 CPU in your Amiga with a 68010 and get a 5-10% improvement in speed, although it may actually be slower in interrupt-intensive applications. At Computer Systems Associates, a 68020/68881 board of their design was plugged into the 68000's socket and apparently promptly tripled the Amiga's speed! (The 68881 is a floating-point co-processor.) This kind of warp-drive doesn't come cheap, though. The bare board alone is $595 (US), and the complete board, with both processors and 256K of static RAM, weighs in at a hefty $1500 (US), more than the purchase price of the Amiga. Still, such a board could really power future CAD/CAM applications at a reasonable price.

Other hardware news . . . Tecmar has raised the prices for their Amiga peripherals ($2000 US for their 20 Mbyte hard-drive) but claim these reflect the inclusion of 'extras' like cables, power supplies, etc . . . Two versions of the Amiga RGB-analog monitor have appeared, the 1070 and the 1080. Rick Sterling of Micro Technic Solutions reports the 1070 to be far superior: better resolution, better colour saturation, a longer persistence phosphor that reduces flicker in the interlaced hi-res mode. Some are opting instead for other monitors, such as the

SONY KV-1311cr, and kludging an adaptor... Latest versions of the graphics chips have added a 'half-bright' mode, enabling 64 colours in the medium-res mode... The digital synthesizer in the Amiga has a theoretical top frequency response of 14 KHz, close to hi-fidelity. Most music software though, changes the sampling rate to change the pitch of the note, effectively limiting the top notes to 7 khz. In addition, there is a hardware filter at the output (which is possible to remove) that cuts out sound above the 7 KHz limit... Larry Miller of FAUG has posted some info from Electronic Arts: "ARCHON works with the joystick in the second mouse/joystick port. **Seven Cities of Gold** requires that you *remove* the mouse and plug your joystick into the *first* mouse/joystick plug."

Finally, because Amiga disk drives do not have their DOS on-board (unlike previous Commodore drives, such as the 1541), hackers are busy adapting generic (read 'cheaper') 3 1/2-inch and 5 1/4-inch drives. One experimenter known only as 'Jay' claims, "Drives for the Amiga must be able to do two things most (the ones I looked at anyway) drives do not: 1) They must be able to send 'Disk Changed'; 2) They must be able to latch the 'Motor' signal." Brian Niessen, one of TPUG's Amiga experts on the TPUG Forums on CompuServe, has confirmed this, having seen a generic 3 1/2-inch drive working with an Amiga. This means that those of you handy with a soldering gun can add your own external drive for less than $200. Film at eleven.

Software News

Turning to the latest action in software... The **Transformer** IBM PC Emulation software is scheduled for release January 15, 1986. Beta-test versions work well with various versions of MS-DOS, including the Kaypro DOS 2000, which can use 3 1/2-inch drives. Test versions worked with IBM-compatible printers. **Lotus 1-2-3**, **Symphony** and **Framework** run as touted, as does PC **Crosstalk**, Panasonic's GWBASIC, and **WordStar 2000**, though the latter is *very* slow. Improvements are expected in the release version... Manx's Aztec C compiler is currently in beta test. William Volk of Aegis reports that **Aegis Draw** runs ten times faster when compiled with Aztec. **Aegis Draw** was originally written with the Lattice C compiler. He says that not only do the floating-point routines run up to 20 times faster, but compile time was five times faster, and the object code was almost half the size... Synapse Software is rumoured to be coming out with an Amiga version of its powerful **SynCalc** spreadsheet sometime in February. It's to be called **CalCraft**... Chang Labs' **Rags to Riches** accounting software is being dubbed 'Lameware' by some because — powerful though it is — it apparently doesn't make use of any of the Amiga's special features: no windows, no mouse-control, no multitasking. Check before you buy, as they say... EA's **Deluxe Paint** is getting rave reviews, not only for its features, but for EA's quick release of upgrades. It looks like the industry will adopt the IFF format developed by EA for graphic files, making insertion of graphics into word processors a reality. **DPaint** and Island Graphics' **Graphicraft** can exchange files in their latest incarnations...

The word from Commodore-Amiga is that AmigaDOS will not be put into ROM — the Writable Control Store stays. This makes upgrading to new versions much easier. 30,000-plus upgrade kits containing v1.1 of **KickStart** and **WorkBench**, and AmigaBasic by MicroSoft, have already been shipped, indicating strong initial sales... By the way, if you've bought v1.0 of Micro-Systems Software's well-received **ONLINE** terminal software, get the free upgrade to v1.1 — seems the VT100 and VT52 emulations don't work on v1.0 because the wrong files were linked during the compilation process... Bob Perez is working on an Amiga version of his remarkable **VMCO** terminal program (see *COMPUTE!*, January 1985). It's going to have all the features he *wanted* to have on the Macintosh version and couldn't... The Unix-like operating system OS/9 is being ported to the Amiga and the Atari ST — it should be available this year... EA has announced that they expect to release **Skyfox** in January, **Arctic Fox** in February, **Adventure Construction Set** and **Deluxe Print** in March, and **Video Construction Set** in April or May... Aegis Development will soon be releasing a full line of art/CAD software, all the way from **Aegis Draw** at $199.99 US to **Aegis Pro Draw**, a professional CAD program with everything **AutoCAD** has, plus anything else they can think of, for less than a $1000.00 US. It will make full use (memory permitting) of AmigaDOS multitasking. You'll be able to dump to printer/plotter or telecommunicate while working on your masterpiece....

AmigaDOS Revisions

There has been some grumbling about the slowness of AmigaDOS. Carl Sassenrath, the man responsible for the specifications for the original Amiga DOS, says Commodore decided instead to buy a 'mature' DOS from Metacomco, sacrificing some efficiency for quicker software development. The following information about the next incarnation of AmigaDOS was posted by Jim Meyers on Compuserve:

> The 'mythical 1.2', if and when it is released, will fix a number of bugs in 1.0 and 1.1, some of which were just caught. Most have to do with the functioning of gadgets. This stuff comes from R.J. Mical, as captured from BIX.
>
> - There's a problem with the locking protocol between Intuition and the Graphics Library: highlighted gadgets are not redrawn properly, and rapid clicking on gadgets will leave them inverted.
> - **MutualExclude** of gadgets does not work yet, but will in 1.2
> - System activation of gadgets will be allowed for (but *not* windows!).
> - Simple columnar text-only menu items might be added.
> - **RawKey** and **SelectDown** don't work now during an active requester. This will change.
> - There is a bug in the calculation of cursor position within the container of a prop gadget which has (just) been corrected.
>
> Please bear in mind that after sharing all this info with us and referring to fixes in 1.2, R.J. then said that 1.2 was strictly a mythical product with no release date nor guarantee of release. I presume that Amiga folk *have* to say things like that to keep people from waiting for the next release of Intuition before writing software, or to protect the company in case the release never happens. All in all, 1.2 sounds to me like a certainty, and some unknown portion of it is already done. I do know that some things which might have made it into 1.1 didn't, because there was a firm 'freeze' date for it.
>
> Jim Meyer

People to look out for on the CompuServe forums: William Volk (Aegis Development), Sheldon Leemon (author of *Mapping the Commodore 64*), Carl Sassenrath (former Amiga designer), Paul Higginbottom (Amiga Product Manager), Michael Reichmann (Batteries Included), Bruce Webster (*BYTE* magazine), Bob Perez (**VMCO**), Russ Wetmore (**HomePak**), Micro Technic Solutions, Electronic Arts, and Micro-Systems Software. And last, but not least, Brian Niessen and Roy Reddy, our Amiga specialists on the TPUG Forums (pcs-116,-156,-155). □

The Electronic Cottage

by Ian A. Wright

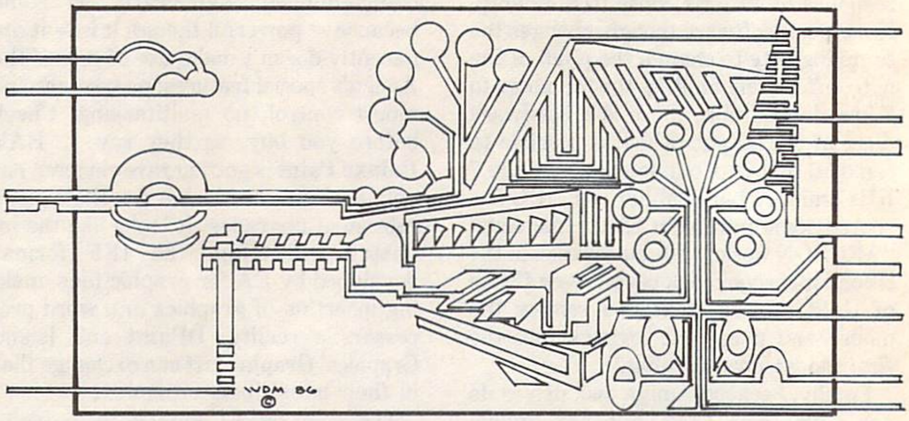
Online databases make the wide world of electronic information available without great expenditures on equipment, and without specialized training. Using personal computers like the C-64, and public database networks like The Source, one-person service businesses can operate out of homes, with truly amazing sophistication. Home computer operators can have just as much access to information as the largest company — and this opens up the possibility of employment via the 'electronic cottage'.

The electronic cottage is the concept of 'networking' computer terminals so that you can do most (if not all) of your work at home, rather than 'going to work'. This kind of job is suitable for many service or 'tertiary' workers like those who work in sales, finance, data management, or similar kinds of activities. Jobs where direct and personal contact is not required are obvious choices for the electronic cottage.

New Jobs

Imagine working at home while providing data services to others. Online material is there, if you know where to look, if you have time to look for it, and if you can manipulate the raw data. Many managers, however, do not have the requisite expertise. Sales managers, for instance, need up-to-date information about trends, and to obtain this, they need current data — often on short notice. A database researcher could contract to provide this kind of material over the telephone lines without leaving home. In fact, one of my ex-students has formed his own company to do inventory control under contract to a local firm, to help pay his schooling expenses.

Using systems like Datapac, iNet 2000 and Envoy 100, home workers could be responsible for running an advertising campaign, operating a mailing service, or coordinating a sales drive. It is possible today. Canadian communication technology equals or surpasses that of the rest of the world — the videotex system called Telidon and the Canadian satellite communications systems are but two examples — so there is a high-tech infrastructure in place. Instant information access and rapid communication are necessary parts of today's business activities — but the huge office building is redundant.



Networking

You don't have to leave the corporate structure to work at home. LANs (Local Area Networks) are based on plugging the limited memory and function of the home computer into the telephone, and using the power of mainframe and mini computers from a remote terminal. Any authorized employee can have access to central files from miles away, to process data rapidly and accurately at home. Similarly, a sales manager can keep track of his staff by calling their computer link. Rather than rounds of endless meetings, colleagues can set up a computer conference through a LAN and use the network to call up reference material such as graphs and charts, while all comments on proposals are automatically recorded. The key is that this could take place over a few days, rather than tying everyone to one conference table or one conference call. All this is possible through the magic of networking.

Some of the larger companies, such as McDonald's, are already allowing some of their staff to work at home through electronic links, and the potential of this kind of employment is enormous. The company doesn't need to invest in a huge corporate building that stands empty while still consuming much-needed resources for almost two thirds of the day. Companies lose millions of dollars through tardiness and illness; much of this could be cut out by the electronic cottage. Managers don't play 'telephone tag' with their clients, because text messages can be left in an electronic 'mailbox'. There's less worry about garbled messages, too, since order messages can be saved as part of a client's file.

Employees gain new tax claims — part of their rent for workspace, some capital costs for equipment and incidental expenses, for example. They no longer need to have a company car or pay daily public transportation costs out of their salaries. Commuting time would be negligible, since there would be only monthly staff meetings and the work week would be spread over seven 24-hour days. No need for mental health days, either — flexible holidays! Child-care expenses can be reduced or cut out entirely, and you won't need those sick days for junior's illness. Office clothing costs can be slashed, because no one cares what clothes you work in at home — as long as you are productive. The savings are tremendous, and the expenses are not.

The greatest benefit will probably come from the return to the old concept of the working family. Father (and/or mother) no longer disappears for eight to ten hours each morning to some unknown and unpleasant place called 'work', to return tired and irritable at the end of the day. The implementation of the electronic cottage will require us to rethink some of the existing structures of our society and review existing roles in the family. Family life was not fragmented until after the industrial revolution — maybe the computer revolution will allow us to return to saner times. □

Punter BBS Commands

Message Numbers and Reference Numbers

Each message has two numbers associated with it: a **Message** Number and a **Reference** Number. The Message Number is the message's current position in the stack of active messages. It is this number you will work with most often. Unlike the Message Number, which changes every time a message is deleted, the Reference Number is unique to each message, and is used to allow you refer back to an old message, regardless of where it may end up in the stack.

There are other uses of this reference that you will soon learn about.

Message Parameters

You will note that, in the Help List (obtained by typing **HELP** at the **Command >** prompt), some commands have asterisks (*) beside them, while others have number symbols (#) beside them. This indicates that the command *can* or *must* have a Message Parameter applied to it (#=MUST; *=CAN, but not needed).

Those with a + beside them have a special syntax that is discussed under the **SCN** and **SEL** commands.

If you want to enter a command without a Message Parameter, simply enter it and press **RETURN**.

If you do want to enter a Message Parameter — or have to — you have two choices: by **Message Number**, or by **Reference Number**. To enter a Message Number, simply follow the command with that number. For example: **DM25**. To enter a Reference Number, follow the command with a dash, then the reference number. For example: **F-1286**. If the Reference Number cannot be found, you will be told **No Reference**, then the closest message to this reference will be used (except in some cases, which will be discussed individually).

Pause and Abort

At any time, you can stop the transmission by pressing the **S** (for 'stop') key. As soon as the current line is finished, the transmission will stop. To start it again, simply press the **C** (for 'continue') key once. Additionally, most functions allow abortion by hitting the **A** key instead of **C**. This **S** to stop, **A** to abort sequence can also be accomplished by sending a **CTRL-A**. In what follows, this sequence will be referred to as the **Abort Sequence**.

Utter Abort

During *any* transmission you may return to **Command >** (or some other places, to be discussed separately) by sending a **CTRL-P**. This sequence will be referred to here as the **Utter Abort**.

Message Forwarding

Any *public* message addressed to you can be forwarded to another user, **ALL**, **SYSOP** or **SUGGESTION BOX**.

Forwarding is done right after reading the message in question. Those functions from which forwarding can be accomplished are: **READ**, **MAIL**, **RALL**, **F#**, **R#**, **R**, **NEW** and **SEL**.

If you are in the **R** (Recall a Specific Message) function, and the prompt says:

(1 — XXX) <H>elp — Message?

Type **F <RETURN>**. If you are in any of the other functions and the prompt says:

<R>eply <F>orward <H>elp — More?

Press **F**. You will now be asked:

Forwarded Msg To
>

Your options are identical to those when sending a message with **E** (Enter a Message). Once a valid name is entered, the new header will be printed. Note: This is a totally new message, separate from the original, but with the old Subject and Text.

Also note that your name appears in the **From** line, but with **(FWD)** after it to indicate this is a forwarded message.

After the message is forwarded, you will be returned to the prompt from which you came.

Replying to a Message

Any message can be replied to directly after reading. Those functions from which reply can be accomplished are: **READ**, **MAIL**, **RALL**, **F#**, **R#**, **R**, **NEW** and **SEL**.

If you are in the **R** (Recall a Specific Message) function, and the prompt says:

(1 — XXX) <H>elp — Message?

Type **R <RETURN>**. If you are in any of the other functions and the prompt says:

<R>eply <F>orward <H>elp — More?

Press **R**. You will now be asked:

Subject of Message?
>

Enter your subject for the message and press **RETURN**. No prompt for a recipient will be seen, since that information is already known. Now enter your message in the same fashion as you would have had you used the **E** command.

Lost Message Recovery

It is possible for you to lose your carrier, causing you to fall off-line. Under most circumstances, this is no great loss — simply call back. But, if entering a long message, that data could be lost for ever.

Fortunately, in **BBS64.3**, if you do fall off-line while working on a message, that message is saved to a special file under your name and a flag is set in your user file to say that there is a lost message pending. When you next sign on, the system will recover the message, put you back in the message editor, and let you complete your interrupted message.

B — Bulletin Section

The Bulletin Section is the place where SEQ files are put for you to read. When entered, you will see the title 'Bulletin Section', followed by a listing of available bulletins, or what you should type to find the bulletin you want to read. If you are in **Expert Mode**, you will not see this list: type **HELP**, **H** or **?** to get it.

The main prompt for this section looks like this:

Bulletin >

At this point, you should enter the name of the item you wish to see. If you enter a name that does not exist, the following error message will appear:

Bulletin Not Found

Most SYSOPs will put **Stop Points** in the files so that you never have to worry about the data scrolling off your screen before you read it.

Although it is possible to stop the printing by pressing **S**, then continue by pressing **C**, the Stop Points make life easier. Stop Point prompts look like this:

Continue?

This is a yes/no question: **Y** or **RETURN** for 'yes', **N** for 'no'. If you answer 'no', you will be returned to the main bulletin prompt.

To exit the Bulletin Section, simply press **RETURN** without entering anything when the **Bulletin >** prompt is on the screen.

The Abort Sequence *and* Utter Abort will both return you to the **Bulletin >** prompt.

Note: Stop points will *not* stop text scrolling if you are in **Continuous Mode**; they will just cause a blank line to be printed.

CAT — List Message Categories

This command simply lists the available Message Categories, which are used *only* in conjuction with the **SEL** and **SCN** functions.

CONT — Continuous Mode Toggle

This function is used in conjunction with the Bulletin Section and message reading. **CONT** is a toggle function: each time you enter it, the mode is either switched on or off, depending upon whether it was on or off in the first place. To use this toggle, simply type **CONT <RETURN>**.

The display will show either: **Continuous Mode** or **Non-Continuous Mode**, depending on the state of the toggle.

When continuous mode is set, the **Stop Points** in the Bulletin files will be ignored. This is useful if you are spooling the bulletins onto a disk file for later printing and don't want a lot of **Continue?**s all over your print-out.

When reading messages, the **<R>eply <F>orward <H>elp — More?** prompt will not appear unless the message just read was addressed to you. Having the prompt appear in this case ensures that you will always be given an opportunity to **Reply** to your own messages. The only way of returning to **Command >** during Continuous Message Reading is to send **CTRL-P** (Utter Abort). The Abort Sequence will merely stop output of the current message and move on to the next.

When you sign off, this mode will be saved in your user file: its state will be re-established automatically when you next sign on.

DM — Delete Message

This function allows you to delete any message sent to or from you. Other messages cannot be deleted by you. Although at sign-off you are given the option of having all messages sent to you deleted, it is still possible to delete selected ones with **DM**, as well as messages you sent to others that you no longer wish to have on the system.

This function requires a Message Parameter. References *must* be found, otherwise you will be prompted for another Message Parameter. If no Message Parameter is given, you will be immediately prompted for one. That prompt looks like this:

```
Msg #       : 25 — Ref 13008
From        : STEVE PUNTER
To          : ALL
Posted      : 1327h on 13-Mar-85
Subject     : A Sample Deletion
Correct?
```

If you answer 'no' to this question, you will be returned to the **Message >** prompt (where you will be taken back to **Command >**, if you simply press **RETURN)**. If you say 'yes', one of two things will happen: if this isn't one of your messages, you will be told so; otherwise, the following message will be seen:

Message Tagged For Deletion

This means exactly what it says. The message has not yet been removed from the system — it has simply been tagged for removal after you sign off. This process is used to save disk usage and user on-time (especially for long-distance callers).

DP — Delete Program

If you wish to delete a program on the download section of the board, you must know its **Deletion Code**. This code was given to the program by the person who uploaded it. Under normal circumstances, this limits the people capable of deleting a program to its sender (and the SYSOP), but the sender may tell another user the Deletion Code via a Private Message (if his or her Security Level allows Private Messages). Type **DP <RETURN>**, or **DP# <RETURN>**. You will prompted with:

Name of File?
>

Enter the full name of the file and press **RETURN**. It will take a few seconds for the BBS to search the directory for the file and then either tell you the file doesn't exist, or show one of the following things. If the program wasn't given a Deletion Code:

Deletion Restricted

If there is a Deletion Code:

Deletion Code?

Enter the code, then **RETURN**. If the code is incorrect, you will be told so and returned to **Command (S:0) >**. If the code is correct, the program will be deleted (which will take from two to thirty seconds, or so).

DUP — Duplex

Normally, the BBS works on **FULL DUPLEX**, which means each character it receives from the user is echoed back to the user. This is the most desirable arrangement, since it lets the user verify that his or her typed characters are getting to the board correctly.

Under some conditions, **FULL DUPLEX** is impractical, and **HALF DUPLEX** must be used (no character echo). **DUPLEX** may be toggled from **FULL** to **HALF**, using the **DUP** command. Type **DUP <RETURN>** and you will see: **Half Duplex** or **Full Duplex**.

E — Enter a Message

This function is accessed either through a direct command at the **Command >** prompt, or via the **Reply** feature of the message-reading functions. This is a very flexible function, as you will soon discover. Before using it, it is wise to fully understand the method by which messages are entered.

Most boards simply have you type in as much on a line as you like (up to about 40 or 80 characters, usually); then press **RETURN**, in order to decide on the format of each line. **BBS64** lets you enter, then justify, your messages, much the same as Steve Punter's **WordPro 64** word processor does.

As you enter your message, it will go onto the screen as expected, *but* once you reach the 31st character, a **RETURN** will automatically be generated, causing your cursor to drop to the beginning of the next line. Do not let this bother you, just continue typing until you get to the end of your paragraph. Then — and only then — press **RETURN**.

To leave a blank line between paragraphs, do *not* simply press **RETURN**, as this is the indicator that your message is complete. Press **SPACE**, then **RETURN**.

As mentioned above, once the message is complete, press **RETURN** on a blank line.

To actually begin entry of a message: type **E <RETURN>** at the **Command >** prompt; press **R** at the **<R>eply <F>orward <H>elp — More?** prompt; or **R <RETURN>**, from the **(1 — XXX) <H>elp — Message?** prompt. You will be asked:

Subject Of Message?
>

Using upper and lower case, type in a subject for this message. The system will only allow you to enter 29 characters, at which point it will just stop accepting characters (except **DEL** and **RETURN**, of course).

Entering only a **RETURN** will abort the function.

If this function was started from **Command >**, you will now be asked:

Send Msg To
>

The reply feature doesn't give this prompt because it already knows to whom the message is to be sent. You have four choices for answers here:

1. A valid user's name. A check of the user file will be made to make sure the name entered is valid.

2. **ALL**, followed by any extra material you like (for example: **ALL BBS'ERS**, or **ALL COMPUTER NUTS**).

3. **SYSOP**, which means System Operator.

4. **SUGGESTION BOX**

Once one of the above is entered, the header of the message will be generated and displayed:

```
Msg #     : 82 — Ref 13100
From      : SYSOP
To        : JOE USER
Posted    : 1351h on 13-Feb-85
Subject   : A Sample Message
```

If this message is *not* to **ALL**, and your Security Level allows Private Messages, you will be prompted with:

Private Message?

Press **Y** or **N**, depending upon what type of message you wish to send. If you answer 'no', or the message is to **ALL**, you will be prompted with:

Category (H=Help) >

For a list of available Message Categories, press the **H** key, otherwise, press the numeric key corresponding to the desired category for this message. Categories are not printed when a user reads the message, but are used in the **SCN** and **SEL** functions to aid in the selection of desired reading material.

You may now start typing your message (according to the rules laid out previously). You have a maximum of 41 lines on which to enter the message, but it can be as long as you like. This is accomplished by appending multiple message parts together on disk. If you exceed the 41-line limit, the following will be printed:

No More Space Left!

If that happens, or you end the message normally (by pressing only **RETURN** on a blank line), it will be formatted and displayed for you. It is formatted to a 38-character line with justification!

Once formatting is complete, you will see:

```
Option:
Send          Insert        Format
List          Delete        Continue
Edit          Abort         Paragraph
Search and Replace:
R/[search]/[replace]/[line #]
>
```

You have now entered the very powerful **Message Editor**. For more information on this editor, see the section on that topic elsewhere in this documentation.

If you want to extend your message beyond its current length, you can. First edit what you have already entered, using the Message Editor, but don't worry about the incomplete paragraph at the 41st line; just leave that alone. Send the message by typing:

S <RETURN>

When the board prompts you with:

Continue Message After Writing?

answer by pressing **Y**. The board will automatically find the end of the *last complete* paragraph and format that to the disk. You will be informed of how much is going to disk by the following message:

Formatting to Line XX

Once the board has finished sending this to disk, it will return and say:

Continue Msg. . .

It will then print out the unformatted lines that make up the still-incomplete paragraph (if any). Continue entering your message as though nothing ever happened. You may repeat this procedure as many times as you like, thus making messages of unlimited size!

EXP — Expert Mode

Expert mode is entered once you have become accustomed to the BBS and how it works. After entering Expert Mode, many of the wordy explanations at the beginning of functions, as well as long option lists, are no longer printed (though all option lists are always available by typing **H, HELP** or **?**, then **RETURN**).

To enter Expert Mode, type **EXP <RETURN>**. You will see:

Expert Mode On

You only have to enter Expert Mode *once*. When you call the function, a flag in your name on the user file will changed so that next time you sign on, you will automatically be in Expert Mode.

If, for some reason, you want to come out of Expert Mode, just type **EXP <RETURN>** again.

G — Goodbye

This function allows you to sign off the system. Once you have typed **G <RETURN>**, you will be prompted with:

QUIT, Are You Sure?

If you accidentally typed **G**, and didn't mean to quit the system, you can say **NO** here, and return safely to **Command >**. If you say **YES**, and there were messages addressed to you, the following will appear:

There were xx messages sent to you,

Do you want them deleted now?

Normally, it is good practice to say **YES**, since it means you won't leave messages you have already read cluttering up the system. The only time you should say **NO** is if your sign-on had to be cut short because of line noise, other more pressing matters, et cetera.

You will now be shown how long you were on-line, and then disconnected:

```
Sign Off At 1428h

Connect Time = 0:25:14
```

LF — Line Feeds

Line Feeds are sent after **RETURN**s by default, but if your terminal is double spacing, you won't want any.

Line Feeds can be toggled on and off by this function. Type **LF <RETURN>** and you will see either: **Line Feeds OFF** or: **Line Feeds ON.**

LOG — System Usage Log

Every time you sign off, your name is stored in a file, along with sign-on and sign-off times, which you can read by typing **LOG <RETURN>**. The Log will appear like this:

```
Starts on: 10-Mar-85

User                        Sign-On/Off

SYSOP                       1125h — 1148h
STEVE PUNTER                1235h — 1310h
JOE USER                    1315h — 1332h
```

The SYSOP will usually choose to keep the log shortened to a 24-hour period so that it doesn't take you a long time to read all of it. Should the log pass through midnight, a **date stamp** will be applied to it. For example:

```
Starts on: 10-Mar-85

User                        Sign-On/Off

SYSOP                       2325h — 2348h
MORRIS BLACK                2352h — 0001h

11-Mar-85
STEVE PUNTER                0035h — 0110h
JOE USER                    0115h — 0132h
JANE SMITH                  0135h — 0201h
```

MINE — List Messages Sent by You

This function gives you a list of message numbers and subjects of all the messages that you have sent that are still on the system. Type **MINE <RETURN>**, and you will be presented with a list something like this:

```
Msg#     Subject

 14      One of My Messages
 67      Here is another from me
```

If this command is followed by a Message Parameter, the listing will start from that message.

NEW — Read New Messages

Every time you sign off, the next reference number is stored, along with your name. Armed with this information, it isn't hard for the BBS to find the first message on the system that you haven't read yet.

Type **NEW <RETURN>**, and the BBS will start a binary search for this first unread message. Once found, the function will continue exactly as the **F** command. If no new messages have been sent (or if they are all private and you aren't allowed to read them), the BBS will return to the **Command >** prompt.

NEXT — Continue Message Reading

At any time, you can stop a message reading function by saying **No** to **<R>eply <F>orward <H>elp — More?**. Once stopped, you can do anything you like (for example, send an answering message, check the log, download a program), then pick up where you left off! To do so, just type **NEXT <RETURN>**.

O — Overview of Messages

If you wish to quickly see who is sending and receiving the messages on the system, Overview will accommodate you.

Messages are read in the reverse direction (most to least recent). Typing **O <RETURN>** will start you from the *very* most recent, while typing **O** followed by a parameter (for example: **O25, O-1598**) will start you from that specific message.

The Overview will look something like this:

```
Msg #       : 55 — Ref 412
From        : SYSOP
To          : JOE USER
Subject     : Just a Test

Msg #       : 54 — Ref 409
From        : STEVE PUNTER
To          : JANE DOE
Subject     : Pretty Common Name Eh?

et cetera
```

QUIT — Sign Off System

This function is exactly the same as **G** but, because it is next to impossible to type **QUIT** by accident, no verification prompt is given.

R — Recall a Specific Message

Messages may be recalled individually by either their Message Number, or Reference Number. To enter this function, type **R <RETURN>**. You will see the following prompt:

(1 — XXX) <H>elp — Message?

XXX is the current number of messages on the system. If you wish to recall a message by its Message Number, simply type in that number, then press **RETURN**. If you wish to recall a message by its Reference Number, put a minus sign before the reference (for example: **-12412**).

It will take slightly longer for a Reference Number Recall, since a binary search of the existing messages has to be undertaken to find your reference. If the reference given does not exist, you will see:

No Reference

The Abort Sequence will return you to **(1 — XXX) <H>elp — Message?**, while Utter Abort will return you to **Command >**.

R — Reverse Message Reading

This function is identical to Forward Message Reading *in every way* – except that the messages are read in the reverse direction instead of forwards. Examples: **R77, R-2300.**

RALL — Read Messages Sent To ALL

This function works like the **F** command, but shows just the messages addressed to all (broadcast messages). You may work this function in two ways: By using it *without* a message parameter, in which case it starts from the *same place* as the **NEW** command; or *with* a message parameter, in which case it will start from that specific message. **RALL** can be aborted and then picked up again, using the **NEXT** command.

Examples: **RALL, RALL55, RALL-2493.**

READ — Read Messages Sent To YOU

If the system has reported that you have messages waiting for you, and you wish to read *only* them, type **READ <RETURN>**. As with **NEW**, you can stop this function at any time to answer a message (or whatever), then pick up where you left off by typing **NEXT <RETURN>**.

MAIL — Read Messages Sent To YOU

This function is exactly the same as **READ** (see above).

S — Summary of Available Messages

This command will give you a list of all available messages, showing only their Message Numbers and Subjects.

If you type just **S <RETURN>**, the list will begin at the most recent message, while if you type **S** followed by a Message Parameter (for example: **S25, S-4752**), the list will begin at that specific message. The list will look something like this:

```
Msg#          Subject
55            Just a Test
54            Pretty Common Name Eh?
et cetera...
```

SEL — Selective Message Reading

With this group of functions (**SEL, SELS, SELF** and **SELT**), you can apply the *same* syntax as with **SCN**, but instead of seeing a summary listing of the matching message, you can actually read them. For instance, if you wanted to read all messages sent from STEVE PUNTER, you would type: **SELF STEVE PUNTER**.

If you stop this function and return to **Command >**, it can be restarted with the **NEXT** command.

See the next section for more detail on function syntax.

SCN — Scan Message Subjects

This function is nearly identical to the **S** (Summary) command, except that it lets you select what you wish to look for. This is accomplished by supplying the function with a Search String. By choosing one of the **SCNF**, **SCNT**, **SCNS**, or **SCN** functions, you may choose to have the search applied to the **FROM** field, the **TO** field, the **SUBJECT** field, or all three simultaneously. Any time the search string is found *anywhere* in the selected field of a message, that message will have its number and subject printed. Additionally, you may select to have only a certain Message Category looked at in conjunction with your search string.

The basic syntax of this command is:

```
SCN [cat#];[search string],[parameter]
```

The Message Parameter is optional. If left out, the search will begin at Message Number 1. The Category Number is also optional. If it is left out, all messages will be searched. The search string itself may be left out, but only if you include a Category Number. This arrangement allows a search for all messages within a certain category.

It is important to remember that if a Category Number is included, you must also put in the semicolon, *even if no search string is given.*

Here are some examples:

Assume you wish to see a list of all the messages *from* the SYSOP (ignore the category, and start from Message Number 1). You would type: **SCNF SYSOP**.

If you wanted to list all the messages *to* the SYSOP, you would type: **SCNT SYSOP**.

Suppose you wanted to list every message in category 2. You would type: **SCN 2;**.

Note that if you left out the semicolon, the BBS would think the **2** was a Search String, rather than a Category Number. Now suppose you wanted to read all the messages about 1541 drives For Sale (assume that is Category 1). You would type: **SCN 1;1541**.

And, to find messages after Number 50 concerning Apples (any category), type: **SCNS APPLE,50**.

By using variations on the above syntax, it is possible to list messages pertaining to just about any subject you wish.

TIME — Show Date and Connect Time

This function will let you see how long you have been signed on to the board this session, and what the date is. Type **TIME <RETURN>** and you will see the date and time in the form:

```
1635h on 03-Apr-85
Connect Time: 0:32:55
```

TOME — List Messages Sent To You

Although **READ** lets you read all messages that were sent to you, you can get a quick list of the message numbers and subjects of all those messages by typing **TOME <RETURN>**.

Following **TOME** with a message parameter will cause it to start at that specific message, rather than the most recent.

An example listing:

```
Msg#          Subject
74            Testing
52            Hi There
25            What's Up Steve
et cetera...
```

U — Examine User List

The User List is the list of all people who have accessed the system. Instead of just reading through all of them (it would take a *very* long time), you give information to tell the BBS where to start the listing. Names are stored in alphabetical order (by first name). Type **U <RETURN>** and you will be prompted with:

```
Starting Pattern Match?
>
```

Enter a string of characters. The BBS will now search the user list for the first reference to these characters (or the next highest string, if they are not found), and begin listing.

For example, let us assume you wanted to start listing from the people whose names started with 'ST': we would simply enter **ST** as our Starting Pattern Match. If you were looking for just a specific user, the best thing to do is to enter the entire name as the pattern match.

Simply pressing **RETURN** for the Starting Pattern Match will cause the listing to start from the beginning.

Uploading and Downloading

The sending of a program *to* the BBS is called UPloading. Taking a program *from* it is called DOWNloading.

To be able to upload or download, you will require a terminal program that supports the Punter transfer protocol. Such terminal programs are available through TPUG for the Commodore PET, C-64 or VIC 20 computers. These programs are public domain, meaning that there are no restrictions on copying them or giving them away.

Directories

BBS64 offers up to ten separate Directories (0 to 9) for uploaded programs. These Directories can be given different Operating Hours — hours when uploading to and downloading from each Directory can take place.

Also defined are types of transmission protocols you are allowed to use in each Directory. The 'old' protocol, C1, or both may be allowed.

Commodore 64 and VIC 20 directories may be restricted to the C1 protocol only, as the old protocol takes much more time to use, thus a user will use up much more connect time. If this is the case, the SYSOP may set up a separate Directory that allows the 'old' protocol, from which you can download public domain C1 terminal programs.

Directories will not necessarily be split by computer type: the SYSOP can decide to use one or all of the ten available directories for anything he or she likes.

Commands and Directory Numbers

To refer to a specific Directory in a command, simply add the Directory Number to the end of the command. For example, **LIST0** will list the contents of Directory 0; **LONG2** will give a long (detailed) listing of Directory 2; **SAVE1** will upload to Directory 1; and **LOAD5** will download from Directory 5.

Every command, except **LIST** and **LONG**, will default to the last accessed Directory, if no Directory Number has been given. If no Directory has yet been accessed, the command will default to Directory 0. **LIST** and **LONG**, without a Directory Number, will simply list all Directories, their Numbers, Names, Hours, and Protocols.

Transmission Protocols

Two separate transmission protocols are available from **BBS64**. The first protocol, generally referred to as the 'old' protocol, is Steve Punter's original. This is the protocol that was used extensively by Commodore PET owners; it was later converted to the Commodore 64 and the VIC 20.

The second protocol, the 'new' Punter protocol (officially known as 'C1'), was written to ensure fast, error-free uploading and downloading. This protocol was not based on any existing protocol, but was designed from the ground up, taking into consideration the problems with all other protocols.

C1 is over twice as fast as the old protocol, and offers extremely accurate transmission, even over very bad telephone connections. Its speed is boosted by its ability to transmit blocks of various sizes: small blocks may be sent under adverse conditions, as opposed to sending full size blocks of 255 bytes. As there is a much greater possibility of getting smaller blocks through, C1 is much faster than other protocols under such conditions.

LIST — List Available Programs

In order to see information on programs you may download, enter **LIST <RETURN>**. If you have not specified a Directory Number, you will receive a listing of all Directories, their Names, Numbers, Operating Hours and Protocols, in a form something like this:

```
# Contents              Hours             Prot
0 Commodore 64          D:2100h-0900h     2
                        U:Unlimited       2
1 VIC 20                D:2100h-0900h     1 2
                        U:0800h-0000h     1 2
2 PET (4000/8000)       D:Unlimited       1 2
                        U:Unlimited       1 2
3 Terminal Programs     D:Unlimited       1 2
                        U:[No Access]     1 2
4 Upload Section        D:[No Access]     1 2
                        U:Unlimited       1 2
Further Information?
```

For further help on using the Load Link Commands, answer **Yes**. Otherwise, just answer **No**.

If you specify a Directory Number when you enter the **LIST** Command, that Directory will be shown to you, in the following format:

```
Directory 0: Commodore 64
ST File Name Sender

P TEST PROGRAM STEVE PUNTER
S A SEQ FILE JOE USER
W WORDPRO FILE JANE DOE
*P PROTECTED PRG JOHN SMITH
*S PROTECTED SEQ SYSOP
*W PROTECTED WP SYSOP
85 Blocks Free
```

Under the **S** column, a * (asterisk) will appear if the file has had a Security Code applied by its sender. In order to download such a file, a user must know its Security Code. Thus, a group of users may transfer files between themselves without worrying about the entire BBS community obtaining them.

Under the **T** column, the type of file is listed. There are three types of files possible: Program (P), Sequential (S), and WordPro (W).

LONG — List With File Descriptions

If **LONG** is used without a Directory Number, it will function in the same manner as **LIST**, and return the same list of available directories. Given *with* a Directory Number, it will list the program name, its size in blocks, upload date and description (supplied at the time of upload by the sender). The **LONG** list will look something like this:

```
File Name       Size        Date Uploaded
SOLITAIRE       33.5        13-Feb-85
- THE CLASSIC CARD GAME OF PATIENCE
SEQ LISTER      1           19-Mar-85
- LIST SEQ FILES TO YOUR SCREEN
85 Blocks Free
```

LOAD — Load a File

Type **LOAD <RETURN>**, or **LOAD# <RETURN>**. You will see:

```
Directory #: Name
Name of File? >
```

Type in the full name of the file. The BBS will now search the Directory for the name given. If it is not found, you will be informed and returned to the **Name of File?** prompt. If it was found, you will be given a quick description of the file (entered by the sender) and told approximately how long a perfect transmission will take:

```
File Description:
>SOME TYPE OF FILE DESCRIPTION
Estimated Transmission Times
Old Protocol:        2.3 Minutes
New Protocol:        1.1 Minutes
Current Block Size:  255
```

Choose Receive Option or press A to ABORT, B to alter Block Size

If you have a terminal program with the 'new' protocol, you may choose to alter Block Size. You would be asked:

Block Size?

After entering a new Block Size (40-255), it will be confirmed:

New Block Size: 200

You are now ready to receive the file (if using one of the supplied terminal programs, press **CLR/HOME** for the menu, then select **Receive a Program**).

If you decide that you don't want to go ahead with the **LOAD**, press **A**: you will be returned to **Command (S:0) >**.

There are numerous ways in which a **LOAD** may end. When the load is complete, one of the following status messages will be sent:

OK

Means the program was transmitted properly.

G/B Signal Lost

Means that your transmission of the **Good or Bad** signal for the current block was somehow lost; transmission has been aborted ('old' protocol only).

Over 5 Tries For a Single Block

Means that the board tried five times, unsuccessfully, to send a block; transmission has been aborted.

System Abort

Means the SYSOP aborted the transmission on you for some reason.

Load Aborted

Means *you* aborted the transmission at some point.

SAVE — Save File to the BBS

Type **SAVE <RETURN>**, or **SAVE# <RETURN>**. You will be prompted for the file name, but also shown how many blocks are available on the chosen Directory:

Directory #: Name
Disk Space Available: XXX Blocks
Name of File? >

Give the name by which you would like your file stored. The BBS will now search the Directory to determine whether or not that name has already been used. If it has, you will be told so and returned to the **Name of File?** prompt. If the name has not been used, you will now be asked:

Security Code (or <RETURN> for General Distribution)? >

As it says, if you just press **RETURN**, the file can be downloaded by anyone (with the proper terminal program of course); while if you do enter something (up to 6 characters), only those persons knowing this code will be able to load the file. You will now see:

Deletion Code? >

If you wish to be able to delete the file in the future, this code is needed. Enter up to 6 characters. If only **RETURN** is entered, only the SYSOP can delete the file. You will now see:

Describe the file in 36 Characters >

Within no more than 36 characters, give a useful description of the file you are uploading. You will now see:

Choose Transmit Option or press A to ABORT

Once the upload is finished, the number of blocks received will be sent back to you so you can determine whether the entire file was sent or not:

If using old protocol:

Done:25 Blocks

If using new protocol:

Done:25 Blocks
6350 Bytes

As with **LOAD**, there are a number of ways that the save can end up. Many of the same conditions exist, but with one extra: **Disk Full** will be returned in the Status Variable (see **STAT**) if the file you were uploading exceeded the available space on the diskette. Should this occur, the file being uploaded will be deleted from the disk and the Directory will be left looking exactly as it did before the upload began.

Note: In both the **LOAD** and **SAVE** functions, a report of how well the transmission is progressing is shown on the screen by a series of dashes and colons. Each dash represents a properly transmitted block, while each colon represents a badly transmitted block. In the new protocol only, a * (asterisk) represents a properly transmitted last block.

You do not need to specify which of the two Punter protocols you are using. **BBS64** will recognize it automatically, then set itself to use that specific protocol.

STAT — File Transfer Status Message

After doing a **LOAD** or a **SAVE**, a message was sent to tell you how the transfer worked. If, for some reason, you didn't get to see this message, or you would like to see it again, simply type **STAT <RETURN>**.

CBM — Change 'Old' Download Timing

Due to a problem in timing between C-64 'old' protocol download software and PET 'old' protocol download software, both require different setups. To change between Standard C-64 and CBM timing, type **CBM <RETURN>**. You will be presented with either:

Standard Timing

or. . .

Old CBM Timing

Aside from these messages, you may quickly verify the timing scheme selected by looking at the command prompt itself. Where **N** is the last referenced directory, it may say:

Command (S:N) >

or. . .

Command (C:N) >

Message-Editor Instructions

The Message Editor is a powerful tool for correcting errors in messages you have just entered. Each command follows a basic syntax that is easily learned and remembered, as only the first letter of the command is needed.

The commands fall into two groups: those with arguments, and those without. Commands with arguments are entered by typing the command's letter followed immediately by the argument. For example, if we wished to edit line 15, we would type **E15.** *Since you may not always remember to type the argument, it will be asked for should you forget. Commands without arguments are even simpler: you just type the command's letter followed by* **RETURN.**

The following sections detail all the Message Editor commands.

L — List

Parameters accompanying Message Editor functions refer to line numbers in the listed form of the message (as typed), not the formatted form. To see the message in its listed form, type **L** followed by the line number at which the list is to start, followed by **RETURN**. Listing can be stopped with any of the abort operations (**CTRL-A**, **CTRL-P**).

A — Abort Message

If you are totally dissatisfied with the message and feel that no form of editing would rescue it, you have the option of totally aborting. Simply type **A <RETURN>**. To prevent accidental aborts, you will be asked:

Are You Sure?

Answer **Yes** or **No**.

C — Continue

Once you have exited the message entering routine, and have had the message formatted for you, you may go back and add to it with the Continue command. Type **C <RETURN>**, and you will be asked:

Continue On Same Paragraph?

If you want the text you enter to be part of the last paragraph in the existing message, answer **Yes**. If you want this new text to be separate from the last paragraph of the existing message, answer **No**. You may now continue as if there had been no interruption.

D — Delete Line

Entire lines can be quickly deleted with this function. Type **D** followed by a range of lines (see **LIST**) to be deleted. For example: **D5** deletes only line 5; **D4-** deletes all lines from 4 onwards; **D-10** deletes all lines up to, and including, line 10; and **D11-16** deletes all lines from 11 to 16 inclusive.

E — Edit Line

This function was added for those used to it from other systems, but Search and Replace is a far easier function to use to fix errors in a line. In the event you would like to retype an entire line, **Edit** is available.

Type **E** followed by the line (see **List**) to be edited. The current line will be listed and you will be given the option of retyping it, or just pressing **RETURN** to leave it alone.

F — Format Message

When you finished typing in your message, it was formatted for you. If you would like to see it formatted again, type **F <RETURN>**.

If you have not yet sent any of the message you are working on to disk (with **Yes** to **Continue Message after Writing?**), typing **F <RETURN>** will result in the immediate formatting of your message. If this is the second, or higher message section, you will first be asked:

Entire Msg?

By answering **No**, you will see only those lines in the current message section. If you answer **Yes**, then you will see your entire message formatted.

I — Insert Mode

You may add text to the middle of an already existing block of text using this command. Type **I** followed by the line before which you wish to insert this new text. The system will prompt with:

Continue On Same Paragraph?

Answer as you did with **Continue**, and proceed to enter new text as though this were the **Continue** command. Once you have finished, hit **RETURN** on a blank line.

P — Paragraph

If you enter **RETURN** on a line, it will contain an End-Of-Paragraph marker. This marker (EOP for short) tells the formatter where to break a paragraph. You can add or remove EOPs from a line with the **Paragraph** function. Type **P** followed by the line at which the EOP is to be altered, followed by **RETURN**.

R — Search And Replace

Spelling errors and other small mistakes can be fixed with **E** (Edit), but that requires typing in the whole line over again, and could result in more mistakes. Search and Replace is much more powerful.

The syntax of Search And Replace (S&R for short) is as follows:

R/[Search]/[Replace]/[Line Number]

For instance, let us assume that line 12 looked like this:

12 and I wouldd like to know just how

We want to get rid of the extra **d** in **would**. Our S&R line will be:

r/wouldd/would/12

Actually, all we would have needed is:

r/dd/d/12

Some caution is necessary. Consider the following line:

27 gett the kettle for me please

If we used **tt** as the search string, *both* references in this line would be found, and that would be incorrect. Always be as specific as necessary, but there is no need to overdo it.

To help you determine if errors (such as the one in the example above) have occurred, the number of substitutions made is told to you. If there were two or more substitutions and you only expected one, check the new line carefully.

S — Send

Once a message has been edited to your liking, you can send it by typing **S <RETURN>**. You will be prompted with:

Continue Message After Writing?

If you ran out of editing space, but still wish to continue with the message after this block is sent to disk, answer **Yes**. If this is the absolute end of your message, answer **No**.

Should you accidentally say **Yes** when you meant **No**, wait until the BBS allows you to continue your message, then just press **RETURN**. The system will verify by saying:

Ok, Closing Off Message □

Since this documentation was written, Steve Punter has added yet more functions to his BBS program, including a **TRACE** feature (for tracing message threads) and a **POLL** feature. Even more interesting is the recent inception of PunterNet, a network of Punter BBSs that Steve hopes may eventually incorporate more than one hundred nodes (individual boards). Look for more information on these developments in upcoming issues of the magazine.

Micro Processes

File Recovery

by John Easton

If you're like me on-line, you sometimes want to get caught up in a hurry, so you set about 'capturing' a whole mess of text at one time, thinking that you'll print it out and digest it later. This weekend I attempted to get caught up with the UNISON system, (which has a 'batch read' capability). I logged on during the football game and (knowing there was a lot to download) hooked my 8250 disk drive to the system and asked for a 'batch read' of everything in my 'Inbox' (their method of letting you know what you haven't read yet). I set everything to spool to disk, and went back to watch the game...

When I regained consciousness, there it was: a full disk, with the last 1900-plus block file marked with the dreaded '*' symbol — meaning that the file wasn't closed properly, and that if I wanted to proceed further with any manipulation of the disk, I ought to validate it — effectively removing that massive file for ever.

Since I hadn't even read the file yet (and apparently had no way to read it, since one can't normally access an improperly-closed file), I got to digging through old issues of *The Transactor* and such for some magic that might help unlock the 'lost' file. (I've since been informed that the *Transactor Anthology* book mentions this 'quirk', but I never got that far.) Sure, I have 'disk doctor'-type programs that will do the job on normal 4040-type disks, but nowhere did I find reference to a job of this size. A call to Jim Butterfield Monday morning at least reassured me that all wasn't lost, and he proceeded to talk of BAMs and tracks and sectors — all very clear to Jim, but less so for me — especially with the current lack of 8250 documentation at hand.

To make a long story short, I *did* read up on all the BAMs and tracks and sectors (and still had to come up with a definitive solution), when another call to Jim (on another matter) prompted him to say: "By the way, you know you don't really have to go to all that trouble with file recovery. Commodore built in a command to do it for you."

As I picked up my chair, Jim went on to explain the magic 'M' command. It seems that if you care to open a file to disk with 'M' instead of 'R' (for read), the disk will just go ahead and read, mindless of any errors it might encounter on the way (maybe that's what 'M' stands for — 'mindless'?). Thus cheered, I sat down and typed something like the following into a **File Recover** program:

```
10 open 2,8,2,"0:oldfile,s,m"
20 open 3,8,3,"1:newfile,s,w"
30 open 4,4
40 :
50 get#2, a$    : rem from disk
60 print#3, a$;: rem to disk
70 print a$;    : rem to screen
80 print#4, a$;: rem to printer
90 :
100 get b$:if b$ <> "a" then 50
110 print#3:close 3
120 print#4:close 4
130 close 2
140 end
```

A word of explanation might be appropriate for line 100. As Jim pointed out, this type of file read doesn't necessarily stop for errors, so you need a way to abort the whole operation, should you encounter garbage. Line 100 lets everything whirl away till you care to enter an 'a' from the keyboard — at which time all is closed down in an orderly fashion, and you now have what was readable from the old file transferred to a new file on a new disk, which you can now read in the normal fashion.

For the curious: yes, it works flawlessly and — thanks to having the foresight to also include a printout — as I worked away on another machine, just listening to the printer allowed me to pick up the end of the file. When the file finally quit, it ran into a 'closed loop', and thus repeated the same block of text several times until my ear picked up the repeated rhythm. I then called the 'abort' function to the rescue.

Time to transfer approximately 1900 blocks of text? About three hours — which isn't far from what it would have taken to print the pound or so of paper that resulted, anyway. But let me tell you, we sure won't need toilet paper in the bathroom for a while! □

Doubling 1526/MPS802 Ribbon Life

by Howard M. Mesick

Catchy title, huh? Sort of hits you in the old wallet.

The expensive 1526 ribbon cartridge is a long box in which a continuous loop of ribbon is fan-folded. The ribbon always runs in the same direction, filling up one end of the cartridge while emptying from the other. It is a plastic band, covered with carbon only on one face.

On each pass, some carbon transfers to the paper. After a few passes, the print becomes noticeably lighter, because most of the carbon is gone. Observe, however, that the used-up portion occupies only the centre third of the band. The top and the bottom are still new.

I haven't figured an easy way to get to the top but, to access the bottom, you need only tilt the front of the cartridge up a little (the front is the part with the two arms from which the ribbon is stretched), so that the print needles strike the untouched bottom portion of the band. If you tip the cartridge too far, you may disengage the feed mechanism and/or lose the lower part of your characters. If the latter happens, turn off the printer, so that you don't excessively wear the lowest needles of the print head. If those wires miss striking the ribbon too long, costly damage could result. You may want to reseat the cartridge normally before trying again.

Once you get used to repositioning the ribbon, it should be easier. Mine stays put fairly well, with only an occasional touch-up nudge. Before printing a long document, run off a test line or two. Examine for cut-off capitals or descenders every few pages. If the cartridge wants to move up or down, a little paper or plastic wrap between the extrusions on the arms, and the slots into which those extrusions fit, may tighten things up. A drop of fast-drying glue also works wonders. Careful! Don't get any on the head transport bands, or lay so much on you can't break the cartridge loose when it's old.

Commodore can't figure out what to call its better dot matrix printer for the VIC/C-64/C-128 line. The first ones were 1526s; then came MPS802s. A local retailer says he's presently getting units marked '1526/MPS802' and 'MP802/1526'. Whatever Commodore calls it, you can save money on the cartridge. □

Nitpicking Through Print Using And Rounding

by Elizabeth Deal

Need a good subject for debate in your neighborhood? Here's one: how to round numbers. Want to know more about **PRINT USING**? Here we go:

Computers that have a formatting command for numbers (**PRINT USING**) usually round the numbers to the nearest decimal point that you specify. If you want to see numbers with two places after a decimal point, you'll code **PRINT USING"##.##"** so that:

- **1.44412** prints as **1.44**
- **2.4494** prints as **2.45**
- **3.44669** prints as **3.45**

Neat and compact.

Let's look at some specific examples of what is involved in rounding. For simplicity, we'll just try to round numbers to the nearest integer. We can probably all agree that:

- **1.987** becomes **2**
- **1.5198** becomes **2**
- **3.5** becomes **4**

But what about 2.5? Some say it should be 3, under the theory of always rounding up. Others claim that 2.5 should be rounded to 2. You can have a raging debate. Such is currently the case in my neighbourhood. Fascinating. Useless. Useful. I don't know. I belong to the group that rounds 2.5 down to 2 because 2 is an even number. Several reasons:

- I have an old Polish math book that says that if what remains after chopping the digits is odd, a 'one' should be added: otherwise, leave it alone.
- A statistics book, while not recommending operations on rounded numbers (bad practice!), says that if you must work with rounded numbers, the cumulative error is minimized if you follow the odd-even rule.
- Pragmatic: I balance my chequebook by estimating. At the end of the month, I'm practically never over a dollar away from what the computers at the bank said.

Time for a demo. We don't need a computer. Let's estimate the sum of four numbers:

Values	Round-up	Odd-even
1.5	2	2
2.5	3	2
3.5	4	4
4.5	5	4
Sums: 12.0	14	12

As can you see, under the odd-even rule, the rounding errors balance out. You can be pretty sure that if the computer printed using this rule, you could safely use the printed values for further math, even though that kind of use is frowned upon in all textbooks.

The sum of the numbers calculated using the round-up rule is a bit 'off'. It's okay for many uses — nothing really bad with that value — but you must *never* do anything with the resulting values, other than look at them. Do not ever try to use the outputs as inputs into another program that might perform calculations on those numbers. You shouldn't even try to use them in plotting things such as pie-charts or histograms, as the total per cent may exceed 100. The cumulative errors just get too large.

It's unfortunate that Commodore chose to use the 'always round up' rule in the Plus/4, B-128/256 and C-128 computers. It seems like an arbitrary decision that could backfire. I can understand the taxation authorities telling you to always round up to the nearest dollar — they have to balance the budget, somehow. But my home computer has nothing to balance . . . □

The ABasiC Dir Command

by Chris Johnson

If your Amiga has only one drive, the **Dir** command in **ABasiC** will only allow you to see the directory of the **Workbench** (or other) disk that you used to boot the system. In order to see the directory of another disk, you have to use the **SHELL** command:

```
SHELL "Dir <dir>"
```

where **<dir>** is the name of the disk you want catalogued.

This also allows you to use the redirection capability in the Command Line Interface (**CLI**). If you wanted to make use of the names of the files on your disk, you could send the directory to a file instead of the screen, then read that file from your program, entering the file names into a string array.

```
SHELL "Dir > DF0:DTY"
```

This will send the directory to a file called **DTY** on the disk in **DF0:**. It gets a little trickier if you want to specify a sub-directory. The name of the sub-directory must follow the redirection command:

```
SHELL "Dir > RAM:DTY DF0:C"
```

This will send the directory to a file in **RAM:** called **DTY**. This can be opened from **ABasiC** in the same way as a disk file, but if you want to use the file names, **List** would be easier to use, since it puts just one file or directory name to a line. With the **List** command, the redirection can come at the end of the line:

```
SHELL "List DF0:C > DF0:Files"
```

To those of us used to working with 64K RAM at most, it may seem strange, but an Amiga with only 256K RAM is somewhat limited. The **SHELL** command may not work, particularly if you have a large program in memory or large files in **RAM:**.□

Fear Of Mice . . .

by Chris Johnson

Do I dislike the 'mouse' because I think the keyboard is a better person-to-computer interface? Or do I dislike it because there is never a large-enough clear space on my (actually quite spacious) desktop? Whatever the reason, I find many tasks require the use of the keyboard, and having to use the mouse can often be an annoyance, rather than a help.

The AmigaDOS Command Line Interface (**CLI**) is a more traditional user interface, with similarities to CP/M or MS-DOS. It is in the **System** drawer on the **Workbench** disk. (It must first be activated by using **Preferences**).

CLI has a number of very powerful commands (see *TPUG Magazine*, December 1985). I often do not want to go through the **Workbench** to get at a facility like **ED**, (which I am using to write this article). Instead, I use a disk that will go straight into the **CLI** when the Amiga asks for the **Workbench** disk at power-up.

To prepare such a disk, use the **Install** command from the **CLI**. However, there will be a problem if you only have one drive. The syntax is **Install DF0:**. Unfortunately, this will only install the disk that is already in the drive. To get around this, the **Install** command must be loaded into **RAM:**. The following sequence will **Install** a disk so that it can be used in place of **Workbench** on power-up:

```
COPY :C/Install to RAM:
```

Replace the **Workbench** disk with the (formatted) disk you want to install:

```
RAM:Install DF0:
```

For this disk to be useful, you will want some of the **CLI** commands on it. These are contained on the **Workbench** disk in the directory :C. The easiest way to do the transfer seems to be to **Copy** the directory into **RAM:**, then **Copy RAM:** to your disk:

```
Copy :C TO RAM:
```

Insert your disk in the drive.

```
RAM:Makedir DF0:C
RAM:Copy RAM: to DF0:C
RAM:Install DF0:
```

Depending on what you are going to be doing, you may also need to copy one or more of the other directories from the **Workbench** disk. The same procedure will work for them.

Using the **CLI** instead of **Workbench** may mean you will have to do a little more housekeeping yourself. **ABasiC**, for example, requires a larger stack than the default of 4000. **Workbench** would take care of this for you; **CLI** does not. In order to run **ABasiC** from a **CLI** window, you must first increase the stack with:

```
Stack 8000
```

Using **Workbench**, it takes about forty seconds from the time the Amiga asks for the **Workbench** disk to the time you can be in **CLI**. Booting directly into **CLI** can be accomplished in ten seconds. □

Library Additions

VIC 20 Disk (V)G2

Presented by Richard Best

TPUG has just issued an all new VIC Library catalog, listing all of the disks and tapes now available. (Many thanks to Anne Gudz for her perserverance. Thanks also to Louise Redgers and Doris Bradley.) Many have never appeared in these pages so, beginning this month, I will be featuring (at least occasionally) back issues in addition to regular monthly disks. This month I would like to introduce you to (V)G2, one of our games compendia.

There's lots of vintage entertainment on this reissue. **Slither** 4 adds barriers to this classic game. A popular puzzle game has been recreated in **Towers Of Hanoi**. Watch out for rocks and flower beds when you play **Lawnmower**. Become a part of Greek mythology by helping **Icarus** to recover lost feathers. **Galactic Code** will let you encode and decode messages by substituting numbers for letters.

Turnip is a lot like **Slither**, except that this worm likes to eat carrots, and sometimes his tail becomes a lit fuse. Pilot your ship through five treacherous caves in **VICQuest**. Watch out for the aliens. **Queen's Plate** brings you the excitement of the track. Up to four may 'play the horses'. **Slither Duel** is more traditional, with two opponent snakes playing against each other. See the psychologist's maze from the inside in **Ratrap**. Use your joystick to race for the cheese.

During **Visiting Hours** at the hospital, you must search out a variety of objects and collect them into one room. Or gather the Rebel Forces together to defeat Darth Vader in **Jedi**. **Nitemare Park** becomes a real nightmare when you encounter villains and obstacles in this deceptively quiet place. Experience the 'final frontier' once again in **Star Trek**. Clear all 16 quadrants of Klingons without resorting to the deadly Corbomite.

Don't forget to send for the new catalog. And keep those programs coming in! □

```
0 vic-20 disk (v)g2
14     "list-me (v)g2"        prg
28     "ratrap8k.v"           prg
28     "rescue8k.v"           prg
35     "jedi8k.v"             prg
51     "star trek16k.v"       prg
18     "visiting hrs8k.v"     prg
26     "nightmare pk8k.v"     prg
11     "slither duel3.v"      prg
3      "slither boot.v"       prg
12     "slither 4.v"          prg
10     "galactic code.v"      prg
12     "turnip boot.v"        prg
11     "turnip main.v"        prg
12     "towers o hanoi.v"     prg
7      "icarus boot.v"        prg
8      "icarus main.v"        prg
10     "lawnmower.v"          prg
10     "vicquest boot.v"      prg
11     "vicquest main.v"      prg
4      "queens plate.v"       prg
343 blocks free.
```

VIC 20 Disk (V)TP

Presented by Richard Best

You may have noticed that issues (V)TM, (V)TN and (V)TO are missing. In order to keep pace with other libraries, we are leaping ahead to (V)TP.

The December disk is, as usual, a marvellous collection of members' output. We have a nice variety of games here, and some very good educational and productivity programs. **T'Bird-Joy** is a fast-moving **Breakout**-style game that should provide hours of fun. **Apple Pick'n** has the player digging for apples and trapping monsters, while music plays throughout. More music, this time polyphonic, is available from **Fugue**. The melodies are stored in arrays. **Planets** you've seen before. This update corrects the position of Pluto.

Geom Plot features a novel approach to graphing a geometric equation, and allows replotting when the graph 'blips' off the screen. **Breakout** is standard, but should be useful for budding game players or game authors. For the statistician in the family, there is **Roll Dice**, which calculates probabilities. **B-B-Bounce** is a unique lesson in animation, in the form of a game skeleton complete with paddle, barriers and bouncing ball.

Still can't figure out metric (or imperial)? **Metrivert** will do it for you, including weights, measures, distances and temperatures. Two small data bases, **Paperboy** and **Chequebook**, will help you keep track of your paper route and your bank account.

Logger is a nicely-written **Donkey Kong**-type game — just right for the younger player. **Grungy Towers** has been updated once again, adding better logic and VIC-oriented screens. **Budgeting** is a powerful little financial manager. This menu-driven wonder should have your books in shape in no time.

QCUG of Belleville has graciously contributed a graphics utility called, simply, **Graphic**. It adds SuperExpander commands to an expanded VIC. Three demo programs are included that show how the commands are used. The routines are much faster than the SX. □

```
0 vic-20 disk (v)tp
10     "list-me (v)t-o/1"    prg
8      "list-me page 2"      prg
15     "t'bird-joy.v5k"      prg
6      "apple boot.v5k"      prg
11     "apple pick/n"        prg
7      "fugue.v"             prg
1      "planets.ins"         prg
12     "geom plot.v"         prg
10     "breakout.v5k"        prg
7      "roll dice.v"         prg
6      "b-b-bounce.v5k"      prg
7      "metrivert.v"         prg
7      "paperboy.v"          prg
9      "chequebook.v"        prg
16     "logger.v5k"          prg
22     "log"                 prg
28     "grungy twrs.v12k"    prg
16     "budgeting.v12k"      prg
9      "graphic.v12k"        prg
2      "canada flag"         prg
3      "plotter"             prg
4      "gr. amer. hero"      prg
451 blocks free.
```

PET Disk (P)TP

Presented by Mike Donegan

Most of this month's PET disk consists of educational programs, including **Enhanced Logic Simulator**, by Robert McDermott, an intriguing digital logic simulation program. This was originally a C-64 program, and requires **PETload 64.p** (included on the disk) to load. **PETload 64.p** will not fix peeks, pokes, sound or graphics stuff.

The other educational programs are part of a set started on the November PET disk, (P)TN. They include **Resonance.p**, a program to help you study resonance and standing waves in

Software order form

NAME ____________________

STREET ADDRESS ____________________

CITY/TOWN/P.O. ____________________

PROV/STATE ____________________ POSTAL/ZIP CODE ____________________

TELEPHONE ____________________ MEMBERSHIP NO. ____________________

TORONTO PET USERS GROUP, 101 Duncan Mill Road, Suite G7, Don Mills, Ontario M3B 1Z3 416-445-4524

disks

To order club disks by mail, send $10.00 for each 4040/2031/1540/1541 disk (4040 format), discount price 5-10 $9.00 each, 11 or more $8.00 each; 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

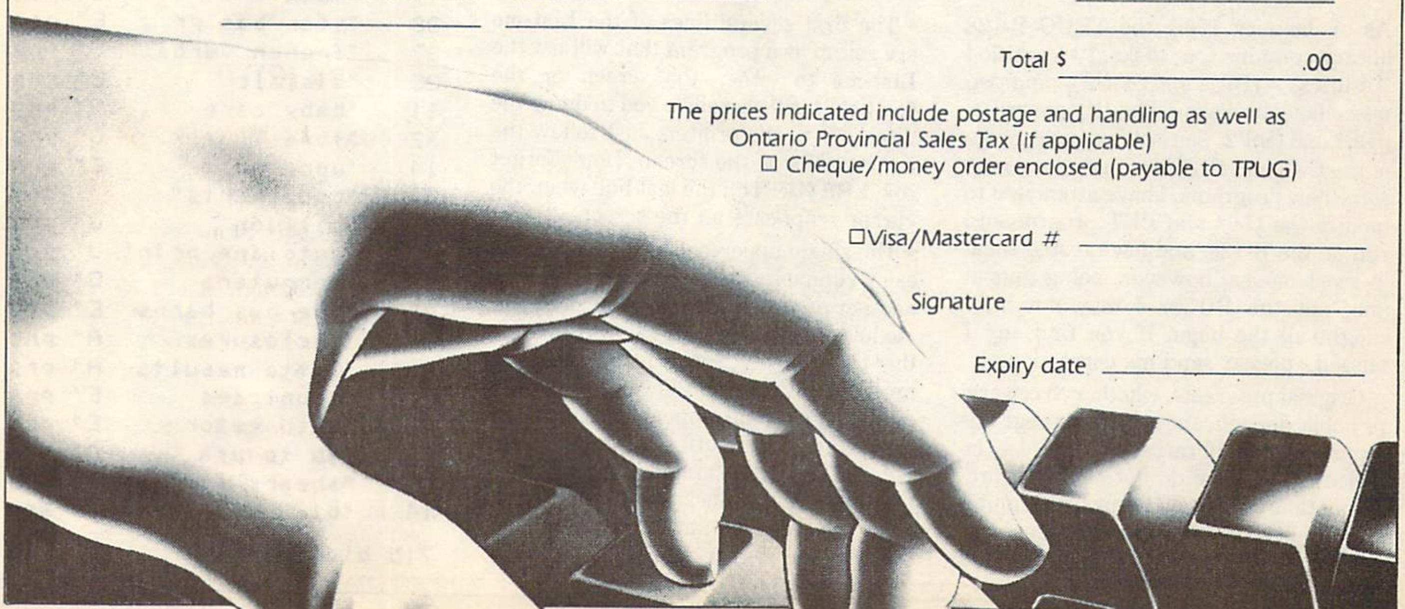
Total $ ______ .00

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 # ____________________

Signature ____________________

Expiry date ____________________



strings and air; **Key Signatures.p**, to teach one to read the musical staff; and **Canadian geog.p**, an aid in learning to identify Canadian geographic locations.

Ascii <--> Petscii, by Ian Wright, is a utility program that converts ASCII files to PETSCII files and back, from a source file on either tape or disk, to your choice of the screen, printer, tape, or disk as the destination. This is also a C-64 program requiring **PETload.p** to load. Another true PET utility is **Disk Check.p**, to look at and check your disk.

Finally, there is a version of Jim Butterfield's popular **Supermon+** for the PET. You will find a documentation file for this monitor on C-64 disk (C)TL. □

```
0 pet disK (p)tp
15    "list-me ptp.1"      prg
82    "logicsim adv.c"     prg
3     "jK flip-flop"       seq
1     "and"                seq
1     "or"                 seq
2     "full adder"         seq
32    "logicsim.c"         prg
56    "resonance.p"        prg
36    "meter reading.p"    prg
30    "forces.p"           prg
58    "chemistry prob.p"   prg
54    "equivalents.p"      prg
54    "equations 2.p"      prg
35    "canadian geog.p"    prg
38    "Key signatures.p"   prg
24    "periodic prop.p"    prg
15    "ascii/petscii.c"    prg
5     "petload 64.p"       prg
14    "supermon+.p"        prg
17    "disK check.p"       prg
92 blocKs free.
```

The B-128 Library

Presented by Paul Aitchison

As of January 1986, the TPUG B-128 library contains four disks, two entitled 'Utilities' — (B)U1 and (B)U2 — and two miscellaneous disks called 'Potpourri' — (B)P1 and (B)P2. The latter are gleanings of the C-64 and PET libraries, along with some new programs. I have attempted to modify the C-64 and PET programs to run on the B-128, and have tested them in most modes; however, being human (me, not the B-128), I may not have caught all the bugs. If you find any I missed, please send me corrections.

Original programs, whether 'freeware' or public domain, and rehashed programs from existing libraries, would also be welcomed. Please do not submit a program that may have copyright limitations, unless the author has given permission, in writing, for it to be distributed as public domain or freeware. (The term 'freeware' describes programs, usually of commercial quality, that the author has placed in public domain libraries, leaving it to the user to determine how much, if anything, should be paid.) Remember, the B-128 is a machine that is no longer in production, and is unlikely to get any further commercial software support, so support will only come from users' groups such as TPUG and CBUG. To further this goal we need your contributions. Even if your program seems frivolous, or unlikely to be of general interest, it may contain useful routines that others may learn from.

If you include documentation with a program you submit (very much appreciated), please include it on the same disk in **Superscript II** file format. Most B-128 owners have access to this fine word processor. We will include your documentation on the disk when it is released, so use care (and the spelling checker) when creating it. We reserve the right to edit submissions for clarity, language and length, but we'll attempt to keep it as complete as possible.

If you request B-128 disks from TPUG, please state your disk format (8050 or 4040). TPUG supplies disks in 4040 format unless otherwise specified, and you will not be able to read them, of course, on an 8050 drive. TPUG disks are *not* copy protected. By the way, it is not good practice to write to disks not formatted by your own disk drive. It is therefore wise to work from backups of your TPUG disks rather than the originals.

The **List-me** and program name conventions for the B-128 disks are slightly different than those of other TPUG libraries.

Here are the differences:

- The first several lines of the **List-me** are set up as a program that will list the **List-me** to either the screen or the printer. It will also allow you to dump the directory to the printer. Just follow the instructions on the screen. Don't forget to hit **RETURN** on the last line when the cursor reappears on the screen.
- The single uppercase letter suffixes of each runnable program (files used by another program, and machine language modules, do not have this suffix) indicate the type of program. Your computer must be in upper and lower case mode, or the suffix will appear as a graphic character. The **U** suffix indicates a computer utility, for example, and the **E** suffix indicates an educational program. A complete list of the suffix meanings is in the **List-me** file on each disk. □

```
0 b-128 disK (b)p2
25    "list-me (b)p2  L" prg
35    "frosty         M" prg
47    "bible clues    E" prg
58    "quiet afternoonD" prg
19    "cnd prov caps  E" prg
25    "early setl     E" prg
27    "n.s. cities    E" prg
19    "n.s. counties  E" prg
17    "days of r life E" prg
30    "stocK exchange G" prg
14    "cinquian       E" prg
26    "drill er/ir/ur E" prg
16    "drill ei/ei    E" prg
18    "drill ou/ow    E" prg
61    "jesu           M" prg
26    "address booK   B" prg
8     "toKenizer      U" prg
33    "mail           B" prg
68    "checKbooK      B" prg
1     "balance"          seq
1     "trcounter"        seq
14    "blocK modifier U" prg
3     "load address   U" prg
5     "christmas      D" prg
5     "square root    E" prg
7     "disK tidier    U" prg
9     "lemonade       G" prg
12    "banKer         G" prg
29    "bach invent    M" prg
2     "anagram helper U" prg
26    "list-me (b)p1  L" prg
18    "high-q         G" prg
4     "bird of death  G" prg
13    "resistor/cap   A" prg
82    "logic simulatorA" prg
32    "basic logic simA" prg
3     "jK flip-flop"     seq
1     "and"              seq
1     "or"               seq
2     "full adder"       seq
34    "ideal mass     E" prg
33    "elect srvc calcA" prg
17    "math           E" prg
38    "star bas pr    E" prg
97    "french verbs   E" prg
25    "disKalc        B" prg
11    "baby care      G" prg
32    "forest walK    G" prg
14    "word test      E" prg
2     "body parts"       seq
7     "million        G" prg
6     "autoline print U" prg
14    "computers      D" prg
42    "the bus barns  E" prg
26    "enclosures     A" prg
20    "lotto results  A" prg
8     "countries      E" prg
4     "math tutor     E" prg
2     "gb to usa      A" prg
49    "shears scoring A" prg
14    "b128 sup'scrpt2"  seq
715 blocKs free.
```

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 $210

C Compiler

A full implementation of K & R standard C, with double precision variables. US $138 Cdn $170

BASIC09

A fully structured, sophisticated semi-compiled (I code) BASIC. See Byte magazine April 1984. . US $110 Cdn $155

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 $90 Cdn $127

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

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 - $11.00/box
DSDD - $14.00/box
(Minimum purchase - 5. Delivery extra.)

P·G

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

VISA MasterCard

Reviews

Vizastar

from Solid State Software

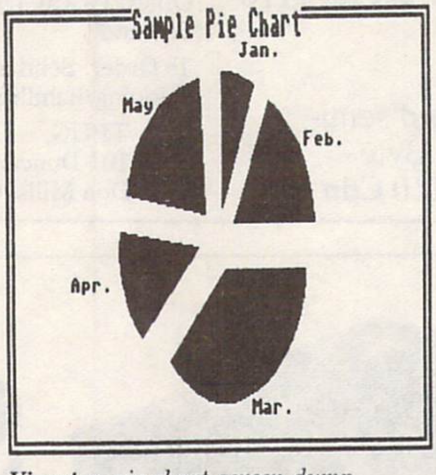
Integrated spreadsheet and database package for Commodore 64, disk drive and printer

Review by Malcolm O'Brien

At the present time, **Lotus 1-2-3** is the most popular software package on the market. Evidently, integrating a spreadsheet with a database and a business graphics program was just what the comptroller ordered. A lot of users are using *nothing* but **1-2-3**! If you have an IBM (or compatible), you can use it too. Of course, after having spent thousands on the IBM, you'll still need to cough up about 500 dollars for a copy of **Lotus**. A hard disk couldn't hurt, and lots and lots of RAM, and maybe an AT....

There *is* another solution, though. You can keep your C-64 (or C-128) and buy **Vizastar**. Programmer Kelvin Lacy set out to produce a program for the C-64 that had the functionality of **Lotus** — quite a task! After more than fifteen months of work, he had the 45 thousand bytes that make up the slickest piece of software you ever saw. **Vizastar** is marvellous! Well thought out, well documented (especially the reference manual), easy to use and fast. In fact, Solid State claims that the **Vizastar** spreadsheet is faster than some spreadsheets running on IBMs! I am not in a position to be able to confirm or deny that claim, but I *can* affirm that **Vizastar**, which is written entirely in machine language, performs all its functions quickly (1541 disk operations notwithstanding).

Like **Lotus**, **Vizastar** has a very simple user interface that cloaks a very sophisticated program. In fact, **Vizastar** is superior to **Lotus** in some ways! Three of my **Lotus** pet peeves are conspicuous in their absence from **Vizastar**. **Vizastar** allows control over automatic page numbering, while **Lotus** resets to Page 1 with each print operation. **Vizastar** will give you a full disk directory on request, but **Lotus** displays only **Lotus** files and bytes free. To change a single character in a **Lotus** database requires loading the file into memory, changing the character and then resaving the file. This is very distressing when the file is a quarter megabyte (or larger!) in size. Mercifully, the **Vizastar** database keeps its records on disk and affords the user complete control over which records, and how many records, will be in memory at any given time. Memory, by the way, amounts to 10K (XL4 version), 14K (XL8 version), or 40K (C-128 version). Although **Lotus** users might think this is far too little free RAM, it can be used very effectively.



Vizastar pie chart screen dump

While we're on the subject of the different versions, I should include a few notes in the matter of peripherals, as this may be of particular importance to you. First, a few quotes from the documentation:

Disk Drives

"Only the 1541 is currently supported, because the other non-Commodore drives are not 100 per cent compatible. One 1541 is required. It will NOT work with the MSD drives, although we will be working with MSD to find a solution. It WILL work correctly with IEEE drives, such as the 4040/8050/8250/1001 if properly connected to the C-64 with a suitable IEEE interface."

"Vizastar can be used with IEEE drives by using the DAMS IEEE cartridge (V3.0 on). When ordering this cartridge ensure that you ask for an extension cartridge socket to be fitted. Vizastar can also be used with INTERPOD available from Cheetah Marketing. INTERPOD interfaces the C-64 to IEEE disk drives and printers [via the serial bus], and also to RS-232 serial printers."

Though no mention is made of the Buscard or the C-Link, it is to be hoped that **Vizastar** will function correctly with these. But be sure! Check with your dealer, or write or call Solid State Software.

Printers

"The pre-set assumption is that you have a Commodore printer connected such as the 1525, MPS-801, or 1526 (with cured timing problems). Vizastar can print out to both Commodore Printers and directly to parallel printers via the user port and a special connecting cable (available from many suppliers and VIZA SOFTWARE. If you have a parallel printer connected to the serial socket via an interface then set a printer type of 'a'. Vizastar will then send out true ASCII characters via the serial socket. If you wish to send information through secondary addresses to your printer or interface this can be done..."

Once again, it is not clear whether you can use a parallel printer connected via the Buscard. The documentation also states that "On Epson compatible printers you are able to print out a full hi-resolution screen copy of your worksheet graphs", and that "you can take a printed copy of a screen display by pressing the Commodore key followed by 3." Regrettably, I had no printer with which to test these features.

The Spreadsheet

The cartridge portion of **Vizastar** contains 4K (XL4 version) or 8K (XL8 version) of memory. On power up, it tells you to press the space bar to load **Vizastar**. If your 1541 is device 9, press shift-space. It takes about two minutes to load, and you are then presented with the familiar row and column layout of a spreadsheet. Pressing tne Commodore key will present you with the command menu. As with **Lotus**, you can cursor (using spacebar) and press **RETURN**, or simply type the first letter of the required command.

The spreadsheet has extensive capabilities: 64 thousand cells, individually variable column widths, alphabetic or numeric sorts, date calculations, two table look-up functions, 32 built-in functions, nine windows onto the sheet. But that's not all. You can merge other worksheets (or parts thereof) to overwrite cells, or to add or subtract with the current cells, and merge word processing files, disk directories or sequential files. You can create bar and line graphs, exploded pie charts or 3-D multibar graphs. You can also set cell display format, set auto or manual recalculation, protect cells, auto skip after cell entry, display formulae or values, and even change screen colors.

When you save your spreadsheet, you also save current cursor location, worksheet file name, active windows, active graphing, 'recalc' and 'skipto' settings, display values or formulae, global format, colour settings, EXEC column/row, all printer menu options and last commands used. Gee, this program is smart! When you reload the file, you're right where you left off.

The Database

Like the spreadsheet, the database could stand alone and still be a front-runner. The disk is divided up into half-sectors referred to as 'slots'. An empty disk (1541/4040 format) has a capacity of 1200 records, though in actual practice **Vizastar** will use some slots for its own purposes. Maximum record size is eight thousand bytes, and maximum number of fields is 64. You can have 15 files per database and **Vizastar** keeps track of which files belong to which database and of your position in the three most recently used files. You design your file layout on the screen and you can have nine screens per record. You can set formats or define formulae for fields, use date calculations and generate reports with selection criteria. You can swap data back and forth from the spreadsheet, import an external file, or export data for other uses such as mail merging with your word processor. Typical record access speed (regardless of file size) is three seconds.

In actual practice, you will run out of memory if you try to use fifteen files with nine screens per file. Better you should use a few files a time, and then switch to the next group of files. The database's flexibility is very satisfying. Working with the spreadsheet and the database, or even external files, is practically seamless.

The EXEC Facility

All of the functions of **Vizastar** can be automated using EXECs. An EXEC is simply a list of keystrokes for **Vizastar** to execute. Tremendous volumes of work can be done with the touch of a key. EXECs allow the use of labels, gotos, prompts, user input, loops and decisions. If you can write programs in BASIC (even simple ones), you can write **Vizastar** EXECs. This is where the real power of the program is. You can even load in a new spreadsheet and the EXEC will continue running, using the EXEC in the newly-loaded sheet!

EXECs can be used to manage data for your own particular application, or simply to spare you the tedium of a large number of simple operations. They can be used to enter data, process it, print reports, or a combination of all three! If you can do it with **Vizastar**, you can get **Vizastar** to do it for you.

You're probably wondering what you *can't* do.You can't use the disk's copy command (**c0:**) to copy your files. There is a program provided to backup your data disks. Validating (collecting) the disk is okay, however, and the command channel is available. You can't name a cell or range of cells, you can't link separate worksheets (directly), and there is no online help facility (hooray!).

By leaving out the on-line help, more program functionality could be included, so be glad it's not there. Definitely *do* take the time to read the manual thoroughly. It's well worth the time it takes. Unfortunately, there are some discrepancies between what's written in the tutorial guide and what actually happens when you follow it. When you get to the staples in the tutorial guide, switch to the manual, and thereafter switch back and forth as necessary while you experiment. You should also probably be prepared to spend a week of evenings with 'dummy' files, finding your way around the program and the manual.

This is serious software with plenty of capability. If you're looking for a spreadsheet, take a look at **Vizastar**. If you're looking for a database, look at **Vizastar**. If you're looking for both, look no further. Take the time to get to know it well, and you will be richly rewarded. This is the best system for storing and manipulating data that I have yet seen. It's fast, easy and powerful. I'm more likely to change computers than to change software. Now if they could just port **Vizastar** to the IBM....

Vizastar: the information processor*, from Solid State Software, 1125 E. Hillsdale Boulevard, Suite 104, Foster City, California 94404, (415) 341-5606. Price: $119.97 (US). Cartridge, manual, copy-protected disk and backup disk. If you purchase* ***Vizastar*** *and are not satisfied, you can return it to the place of purchase within fifteen days and receive a full refund.* □

1541 User's Guide: A Complete Guide to Commodore's 1541 Disk Drive

by Dr. Gerald Neufeld

from Datamost

Review by Ian A. Wright

Copyright © 1985 Ian A. Wright

Dr. Neufeld's latest book offers a soup-to-nuts look at the famous (or infamous) Commodore 1541 disk drive. Although the book starts out at the level of how to unpack and plug the 1541 into your system, it rapidly accelerates into 'useful stuff' like explaining the working of each disk command. This book is full of information — for example, it offers *four* different ways to read the error channel in immediate mode (supposedly not possible!). Another example shows how to save a program as a sequential file *and* how to load it back! There are lots of hints, notes, warnings and explanations in the thirteen chapters, five appendices, and over four hundred pages of text.

The *1541 User's Guide* is easier to read than *Inside Commodore DOS*, co-authored by the same person, but it is still not a book for beginners. The *1541 User's Guide* is especially valuable for the intermediate programmer who wants to fix or find out about disk drives — or for the advanced programmer who wants to brush up on the 'position bug' in DOS 2.6 relative files.

There are over fifty tutorial programs to type in, ranging from a complete self-modifying mail list to a program that will let you align your drive. I think that anyone involved in disk manipulation would want to use some of the file repair and checking programs — and all disk users could learn from the material that deals with disk drive maintenance.

Although not cheap at $25.95 (Cdn.), the *1541 User's Guide* should be on the bookshelf of any VIC 20 or Commodore 64 owner who has a 1541 drive. The first time you successfully resurrect a damaged program by following the step-by-step instructions, the book pays for itself. □

Dot's Nice...

by Paul Blair

As the number of Commodore computers has increased, so have sales of support devices. Many users who started out with cassette units have moved up to disk drives. Others, whose needs go a bit further still, are shopping around for printers to complete their systems.

Why buy a printer? The reasons will vary with every user — it may be word processing for school, university or business; spreadsheets, graphics; or, like me, you might find it easier to edit BASIC programs on paper, rather than on the screen.

Given that Commodore themselves sell 'house brand' printers, it may seem surprising that many prospective purchasers have opted to buy non-Commodore printers. Not only that, but they have at the same time elected to pay out some quite pocket-bending prices for the privilege.

The reason for these choices is easy to identify: there are many more companies making or cloning high-quality printers these days, and interfaces are becoming cheaper and more versatile. It has to be remembered that most of these printers cost more, sometimes require more complex techniques, and don't work with all software. However, since the printers offered by Commodore have had a sad and sorry tale of interminable difficulties, needing a string of revisions to be 'fixed', what option did they have?

Now a sensible option *does* exist. It's the new EPSON Homewriter 10, a 'plug in and go' printer specifically designed to work with the C-16, C-64, Plus/4 and C-128 computers from Commodore. Through the courtesy of Epson Australia, I recently had the chance to review a prototype model. (In Australia, the machine goes under the name 'GX80'.)

Epson and Commodore are not strangers to each other. For some years, Epson supplied the mechanisms for Commodore's main printer lines. I still own a 1978 vintage CBM 3022 printer, upon which I make heavy demands. It has never put a foot wrong. I could still buy spare parts for it from Epson, if it did have a spasm. More recently, the Epson 100 Model was chosen by IBM as the main printer for the PC. Epson probably held up to fifty per cent of the dot matrix printer market for a while, which is no mean feat. To put it simply, Epson has a long and successful history of producing printers that are held in high renown.

The new printer aims to fill the gap left by withdrawal of the Commodore MPS801 from the market. Epson set out to produce an 801 clone, but have gone further than that, because they have included additional print modes — including NLQ.

Some explanations would be in order here. Dot matrix printing is created by moving a 'print head' across the page, firing a series of pins at an ink ribbon, which leaves an impression on the paper behind it. The pins are fired in rapid succession, and the effect is to put enough ink dots onto the paper to make readable characters. The width of the characters, the closeness of the dots when printed, and the shape of the pins themselves all add to (or subtract from) the final print quality. As a general rule, the more pins in the print head, the greater the printing clarity and quality to be obtained. The Homewriter 10 has nine wires, which permits very clean printer characters, as well as very refined graphics.

Let me describe the typical print modes that are in use:

Normal print: the base case. Most dot printers print ten characters per inch (no metrics here yet). Thus, 80 columns of print will use 8 inches of paper width.

Compressed print: useful for squeezing more characters onto a line. The more usual spacings fall between 12 and 17 characters per inch. Epson have chosen to use 17, so 8 inches of print will give 136 characters across a page.

Double strike: doing it twice. The printer makes one pass across the line; then returns to its starting position (left or right), and prints the line again. This gives darker print, but there is a corresponding drop in output speed.

Emphasized print: doing it twice, but with a minute movement of the print head between strikes. This darkens and slightly thickens the printed character. In this mode, the print head makes only one pass across the page. Print speed is only slightly reduced from normal.

NLQ: the new buzz words, 'near letter quality'. This is an attempt to permit dot printers to be used for correspondence by producing characters that closely approximate typewriter quality. This mode also requires two distinct passes of the print head to give the requisite fine print output.

Enlarged print: extra width characters, usually double normal print, created on one pass of the print head.

Reversed print: white characters printed on a black ground, just as you can get on your computer screen.

Bit mode: by sending the appropriate information to the print head, this mode allows virtually any pattern or design of dots to be printed.

I said earlier that the Homewriter 10 was a clone of the MPS801. That was not quite right. The 801 provided only normal, enlarged, bit and reversed print modes. The Homewriter 10 provides *all* the modes described above, greatly expanding the versatility of the printer. And it is possible to combine modes, so that you can print double strike and emphasis together. Or you can combine compressed print with enlarged print, to yield a pitch of 8.5 characters to the inch.

In normal print mode, the Homewriter 10 operates bidirectionally. All the other modes operate unidirectionally.

Physically . . .

The Homewriter 10 is a small, neat unit housed in a plastic casing with a steel base plate to provide rigidity. There is only one removable cover — at the front — to expose the print head carriage for ribbon cartridge removal/replacement and periodic cleaning. All other controls and switches are externally accessible.

As supplied, the printer is friction feed, so you may use cut paper (letterheads, for example) if you wish. A cut sheet feeder is likely to be available. A tractor feed unit can be purchased if you use continuous stationery. This unit readily clips and unclips from the top of the printer in about two seconds. The only other external controls are a paper release lever, and three press panel switches on the top front right panel.

The panel also contains some idiot lights to tell you about some non-useful functions (like 'paper out' — as if you need to have a light to tell you). These switches have multiple functions. The ON LINE (OL) switch applies and removes power, as required. The LF and FF switches give line feeds and form feeds. In combination, the OL and FF switches permit print mode selection (normal, compressed, NLQ, and so on).

At the rear, hidden by the feed paper, are mini-switches that duplicate the OL/FF controls on the top panel, plus a few more — Device 4/5 selection, slashed/plain zero character, the 'paper out' buzzer mute, and page length (11 or 12 inch only). These are some of the more useful control functions.

The printer is connected to the computer (or, more likely, the disk drive) by a serial cable that terminates in a plug-in interface. The interface slides into a deep slot at the rear left of the printer, and requires no further connections. This unit apparently contains the software necessary to provide Commodore compatibility, plus the character set. The sort of graphics characters that you see on your screen will be printed out with your listing, eliminating those '[CLR]' and '[WHT]' (and so on) translations that have become a (forced) standard. There is no buffer in the cartridge, and it is obvious that, as the C-64 can deliver characters quite quickly, the interface does slow things down a bit. Still, at 100 cps, the printer is no slouch.

Printer ribbons come in clip-in cartridges, and can be fitted without getting mucky fingers. They seem to contain heaps of ribbon, and should be good for a lot of printing. The ribbon is arranged as a Mobius strip, so it will alternate the impact area with each lap of the ribbon.

Control codes

In this department, the Homewriter 10 matches the 801 perfectly. These control codes are used from software to take care of case switching (upper case/graphics, or upper case/lower case mode), expanded print on or off, print head positioning, line spacing (6 or 9 lines per vertical inch), horizontal positioning (tab), reverse on/off, bit mode and bit image repeat. Having made so many other improvements over the 801, it was disappointing that double strike, emphasized, compressed and NLQ modes could not be controlled from software. And the 6/9 lines per inch restriction prevents production of double spaced drafts.

What does it feel like?

Not to put too fine a point on it — very nice. There is no feeling like being able to plug in a printer, turn it on and literally forget it. Once I remembered the bonus gift of seria. bus lockup that is free with every Commodore 64, and didn't switch the Epson on until I needed it, the printer and I had a pleasant, albeit brief, time together. I would be happy to have one permanently. It has very tidy print, with well-formed characters and symbols; and its multimode capabilities and versatility with different Commodore computers is good.

Some minor problems

A couple of things bugged me, but the sample unit was not a final production model, and some of these things will surely be fixed by then.

Noise — the slim plastic housing leaves little space for sound-deadening material. I found it hard to hold a phone conversation while the printer was biffing away two metres from me.

The paper bail (the bar plus rollers that keeps the paper pressed against the main roller) has no lever. This meant fumbling into the innards, particularly during paper loading.

A stick-on 'ruler' on the front panel showing print spacings would be useful. The paper bail has small notches every ten standard characters, but as the printer can also print at 8.5 and 17 characters per inch, they were not universal.

These are not serious problems, but they do slightly reduce the pleasure of using the printer. I hope Epson can build in software control of character selection in the near future.

Near letter quality

This is the big bonus. Not everyone can afford a daisy wheel printer, or a daisy wheel printer for correspondence and a dot matrix printer for programs and drafts. Many new printers offer NLQ to make their products more widely appealing, and the marketplace has responded enthusiastically by buying in big numbers. No doubt Epson hopes to capture the attention of some of those buyers.

As noted earlier, NLQ makes two passes to provide a printing dot density that makes each character stroke or shape smoother. If you peer closely at the printed characters, you will observe that character formation looks rather rough. Those characters using mainly vertical and/or horizontal lines are fine. The critical test is sloping lines, and there is some jaggedness evident under a glass, or even to the naked eye. By and large, that small criticism would not be important if Epson was the only printer on the market offering NLQ. But there are printers that do print smoother-sloping lines, and prospective buyers (whose thirst for NLQ is inbreasing every day) can be expected to notice.

The Homewriter 10 in NLQ mode produces characters that are a little larger than normal typewriter character size. My old eyes found this quite pleasant, but it has been commented on by people to whom I showed print samples. And it emphasizes a bit more the NLQ's slightly uneven print.

Graphics

I tried both **Doodle** and **The Print Shop** and came up with some pretty pages of banners, moire patterns and birthday cards. **Doodle** was quick, **The Print Shop** terribly slow (a function of the software). I tried to print double strike to get better clarity, but the switch settings seemed to get overridden by the software. At first, I thought the printer was being reset by the program, but a few tests proved otherwise.

I should mention the extreme accuracy of the printer with these programs. A single line border came out as a straight line, not a series of disjointed graphics characters. Well done!

The bottom line

The Epson will be popular because it plugs straight in, requires no set up, wedges or hang-on doodads, and will run with any software that recognizes Commodore printers (and that's most everything on the market that uses a printer).

The small business market will certainly look at the Homewriter 10, because of its no-fuss attributes. People new to computing will also take a good long look, because the Homewriter 10 overcomes the major hassles of hooking up a Commodore computer to a printer. The middle ground printer is difficult to assess. The lack of quality printers in this range has forced users to learn to live with Centronics printers, and many will remain committed to that standard. But they should have a look at the Homewriter 10, because many of its features will appeal to them, too. I know I would consider one for myself.

And a thank you . . .

Last, but not least, a 'thank you' to the folks at Epson Australia, who arranged the loan of the prototype. It takes courage to let a 'warts and all' preliminary model go out for review, because there will always be changes before product release. □

Indiana Jones in the Lost Kingdom

from Mindscape

Arcade-style game
for one or two players
for Commodore 64
$29.95 (US)

Review by Robert J. Sodaro

A fellow reviewer once remarked that a good game was one he would play after the review was written. For me, **Indiana Jones in the Lost Kingdom** is one of those games.

This is Indy's second entry into the software market. The first — a stillborn Atari 2600 game — failed, due to a lack of rules and instructions, coupled with poor gameplay. Mindscape's version suffers only the former affliction, and should succeed on its superior execution and production values.

Designed to combine the action of arcade hand/eye coordination with the elements of an interactive strategy game, **Indiana Jones** is an attempt to provide interest even after hours of play. In this aspect, the programmer succeeded. However, in leaving out the instructions, he left out perhaps the most crucial element.

In place of rules, the following information is offered: famed archaeologist Indiana Jones must find an ancient artifact from a lost kingdom, worth an incalculable fortune. Using the sparse clues provided, and armed with only a mystical cane, Indy wends his way through the hazards of six rooms in his quest to find the treasure before his archrival, Ivar Reiss.

Perils include vampire bats, carnivorous monsters, killer snowflakes, twisting mazes and treacherous cliffs. The game provides a set of clues for each room, as well as a 'hint hotline' phone number, should people *really* get stuck.

The clues are straightforward enough, but it will take even experienced gamers quite some time to determine the action in each room. For example, one room requires you to scale steep cliffs while avoiding bats. Another has you cavorting around snowcapped mountains dodging killer snowflakes, while a third involves an underground maze with hidden horrors and mysterious passages. Joystick movement differs in each, and requires gamers to 'relearn' how to play. (An example: the fire button works in some rooms but not in others.)

In keeping with the tone of the game, I won't reveal the rules — except to say that, once the logic of gameplay is determined, it's easy enough to discover what's required in each room. The trick is to determine *how* to perform the various tasks without killing your seven men.

The lure of the Lost Kingdom is strong, but having to play 'blind' hours on end seems to pose unnecessary difficulties. 'No documentation' may have sounded good in the planning stages, but seems to have wound up as an excuse not to write engaging and coherent instructions. Still, if you have the tenacity to muddle through, there is much enjoyment to be found while accompanying Indy on his continued adventures.

Mindscape Inc., 3444 Dundee Road, Northbrook, Illinois 60062. □

Sea Voyagers

from CBS Software

Educational program
for Commodore 64

Review by Ian A. Wright

Copyright © 1985 Ian A. Wright

Sea Voyagers is an educational program designed to teach the user about the explorers of the New World, and their exploits. Baffin, Tasman, Cabot and twenty-seven other explorers — ranging from the well-known (but wrongly-named) Columbus to the lesser-known (but interesting) Jacob Roggeveen — are presented in the program's varied activities.

Each activity is a learning game, and can be used by students from grade four and up.

The package

The program comes in a neat plastic bookcase with a twenty-seven page manual, a disk, and a keyboard overlay of soft plastic. The manual has instructions on loading and playing the games, as well as tips on winning them.

Fourteen pages of 'Voyager Sketches' covers the facts about each of the thirty New World explorers of the fifteenth to eighteenth centuries. An explanatory section about earlier and later explorers, the reasons for exploring, the difficulties explorers faced, the improvements that resulted from their journeys, and a bibliography with twenty-three references round out one of the best software manuals I've seen.

The reviewers

The program was tested on a friend's two boys, aged eight and ten. Both had little knowledge of the New World explorers before they tried out **Sea Voyagers**, beyond knowing Columbus' name. They now know the names of most of the explorers (although they may not be able to pronounce them correctly!) and they know one or two facts about each of them. They also know which country each explorer sailed from, where he went, and the approximate year of his voyage.

Both the graphics and the sound of **Sea Voyagers** are excellent. The 'Explorer Profiles' plot their information on a world map focused on North America. Some of the world land mass (India, for example) is missing from this map, but that results from the need to focus the map on the New World.

Sea Voyagers could be improved by adding a screen dump of the 'Explorer Profiles' but, considering the diversity of printers and their incompatibilities, it is easy to understand why this feature has not been implemented.

The boys would also like to have the option of additional explorers (like **Print Shop**'s extra graphics disks) to expand the game from its current North American bias. They really appreciated the keyboard overlay, however, because it simplified the running of each of the activities in **Sea Voyagers**, letting them concentrate on the game.

The activities

Explorer Profile: Provides the facts about each of the thirty explorers and their voyages, shown on a map of the world.

Who Goes There?: Guess a mystery explorer from up to five facts, given one at a time. The more facts you need, the fewer points you gain.

Explorer Mix: Gives three facts and three explorers that are scrambled. You unscramble the information to receive points.

Explorer Match: This is like a poker game. The suits in the easy version are the countries that the explorers sailed to find. The centuries when the voyages took place, the regions explored, or the countries sailed for, are the suits in the harder version of this game.

You can play against other people, individually or in teams, or play against the computer. Be careful of the computer opponent — it's good! Overall rating — very good to excellent. □

HRT Super-Res Graphics Board

from High Res Technologies

An internally-mounted, high-resolution graphics card for PETs and SuperPETs

Review by Tom Stiff

In my work, I accumulate a lot of data; data that must eventually be analysed. As anyone who has had to handle large amounts of data will attest, the best way to begin to understand the meaning of large blocks of related data is to display it graphically. The old adage "a picture is worth a thousand words" never held more truth than in the field of data analysis.

In a single night's observing at the York University Observatory, I can easily accumulate more than a megabyte of data in the form of digitized one dimensional stellar spectra, or a few dozen megabytes in the form of two dimensional digitized astronomical images. Producing a hardcopy of these images is very expensive and time consuming. A fast graphics terminal seemed to be the best solution, and I began a search for a PET or SuperPET graphics package.

Months of letter writing, phoning and searching through back issues of every Commodore-related magazine failed to produce any satisfactory results. In the Spring of 1984, there were three manufacturers of hi-res boards, all for 65xx based machines (designed for PETs, but not for SuperPETs); all were priced over 600 dollars (US); and none offered a screen resolution any better than the C-64.

Each system also had serious design flaws that made them unsuitable. One used up a lot of the PET's memory, another required major hardware modifications and yet another limited the PET's capabilities by redefining some of the PET's BASIC keywords to incorporate graphics commands.

At the 1984 Annual TPUG meeting, I discovered High Res Technology's booth, and described my graphics needs to Dan Deconinck. He seemed optimistic about designing a suitable graphics board. That summer, Dan contacted Avygdor Moise (of OS-9 fame, and also author of **PETCOM**) at York University, for details about the 6809 side of the SuperPET. Since my graphics needs were relatively modest, Dan also asked Avy for additional ideas and suggestions. Gradually, a prototype graphics card began to emerge.

During the next year, two prototype boards were produced. Each was demonstrated at a meeting of the SuperPET User's Group, and each time ideas for enhancements were solicited. Ideas thrashed around during these meetings led to further board revisions and improvements. In the early spring of 1985, HRT felt that their card was ready to 'field test'. Accordingly, they installed their graphics card in my SuperPET.

The graphics card was fully transparent and did not interfere with any PET functions, nor did the card use any of the PET's memory. In other words, with the card installed, I was totally unaware of its existence. I could tack graphics sub-routines onto any existing program and the program would run perfectly!

I have been using an early version of an HRT graphics card for about ten months now, and I am extremely pleased with its operation. I use the graphics capabilities of my SuperPET to analyse stellar spectra. A typical spectrum (called a frame) is 500 pixels in length. The spectra are taken with a silicon-intensified television videcon (a sort of fancy digital TV system). All of the observatory's instrumentation, by the way, is controlled by an ordinary PET 2001.

The data are written onto a floppy disk for temporary storage, then handed over to a VAX 8600 for large-scale 'number-crunching'. I then download the reduced data from the VAX to my SuperPET, for graphics display. The graphics program I have written is entirely in BASIC, and has been compiled, using **PETSpeed**.

It takes about thirty seconds to create a full screen image consisting of over 600 line segments. Most of this time is taken by the SuperPET, to execute an auto-scaling subroutine to a VAX 780 system using a VT100 terminal and operating with normal daytime user-loading. To generate an equivalent hard copy graphics display takes about twenty minutes, using an H-P plotter! Furthermore, the HRT card allows me to overlay an infinite number of frames for comparison if I need to do so, or to simply display them, one at a time. The images can be scrolled off the screen, if I wish to plot more data; and instantly scrolled back, if I want to view them again.

With the birth of Super-OS/9, additional enhancements were made last summer to increase the graphics board's capabilities, and to further increase the screen resolution. Screen resolution is 700 pixels on 80 column PETs, and 640 by 200 on 40 column PETs. But — and this is a nice feature — the total resolution is 1024 by 512 pixels, and the screen can be scrolled in all directions.

The graphics card is easily installed into any PET or SuperPET. It requires no external power supply, no soldering, and no other hardware modifications. The board simply plugs into the main board's 6502 slot, and the 6502 chip is moved to the graphics board. It works equally well on either the 6502 side or 6809 side of the SuperPET and — as if that weren't enough — it is perfectly compatible with OS-9. The icing on the cake is that OS-9 users can also use the graphics card's memory as a 64K RAM disk.

The main problem with this card is that software is scarce. You will have to write your own — in machine language, if you want it to be fast. I have a feeling that this will not be a problem for long, since there are several users that I personally know of (and probably many others) that are already developing graphics utilities that will be placed in the public domain, via TPUG.

I have also heard rumours that PH.D. Associates will be marketing a version of **PETCOM** which will support VT100 and maybe VT240 graphics with HRT's graphics board. Perhaps this upgrade will also be available to registered **PETCOM** users for a modest fee. (Are you listening PH.D. Associates?)

This card is the finest and best designed piece of graphics hardware on the market today for any microcomputer — and at any price. This little card has many excellent features and, in some areas, it out-does the illustrious Amiga. For example, I use the graphics card to produce a 'plotting window' on the top two thirds of the screen while I simultaneously use the bottom third of the screen to edit programs, as well as to display numerical results as they are being calculated. In fact, the normal text screen can overlay the graphics screen. I can choose to erase either the text or graphics, or both.

If you want to breathe new life into your old PET and re-kindle some of the enthusiasm you had the day you first lifted it lovingly out of its styrofoam cradle, this addition might be just what you're looking for.

The HRT Super-Res Graphics Board, *from High Res Technologies, 16 EngishIvy Way, Toronto M2H 3M4. Price $200.00 (US), $225 (Cdn).* □

Karateka
from Broderbund
Graphics adventure game for the C-64

Review by Ajay Jindal

Returning home after years of study under a master of karate, you find your village burned to the ground. Your friends and family are scattered, your bride-to-be stolen by Akuma, the warlord whose oppressive shadow has darkened your village since before you were born...

This is the scenario you find in **Karateka**. Your mission is to penetrate Akuma's mountain fortress, to rescue your beloved Princess Mariko. The only weapon you have is your knowledge of karate. You must defeat Akuma's warriors, and ultimately Akuma himself, to achieve your goal.

As the game starts, you have just climbed to the summit of the mountain upon which Akuma's fortress is situated. There you meet the first of the palace guards.

The graphics of this program are stunning. The background scenery is in a 3-D perspective, and all the figures on the screen move very smoothly — as well as casting shadows, too! These graphic effects give the game a movie-like quality. If you ever want to show off your computer's abilities to your friends, this is the program with which to do it.

You control your man through either keyboard or joystick. You can punch, kick, run, bow and advance. The initial response is quite quick, but the actual movement is slow. The program has a 'joystick buffer', so that if you keep pressing the button while the current move is being completed, it will all be remembered, and the moves executed. All in all, the controls aren't difficult to master, and there aren't that many to remember, anyway.

After defeating an opponent, you must run as far you as can into the fortress, until the next warrior blocks your path. The warriors are supposed to get tougher as you go. I have found that this just means that they can take more hits. It is easy to defeat them after your sixth game because they move in patterns, and aren't very 'intelligent'. As you near the end, you must fight Akuma's eagle, then Akuma himself (a little tougher than his men). If you kill them both, you get to hug Princess Mariko. That's it. No next level. No score. Not even a time or rating. Just a message saying that there will be a next time. The end is very disappointing.

This game offers very little challenge. There are a few easy puzzles along the way, but once you have figured them all out, there is no motivation to play again because there is no way to measure your success. You either finish, or you don't.

At first glance, this seems like a great game, but I got bored with it quickly. Be wary before you lay out the cash for this one. □

Adding Power To Your Commodore 64
by Vahe Guzelimian and Steve Cates
Book ($14.95 US) and support disk ($15.00 US) for Commodore 64

Review by Dave Powell

Another of the 'Everything you wanted to know about the C-64' books? Yes, but with a twist — this one leaves out most of the stuff you *didn't* want to know! Buying this book is better than trying to organize every tip given over the last three years in *TPUG Magazine* (and maybe several others) in a logical order. It goes beyond the early attempts — nothing more than rewrites of the Commodore reference books — and gives clear explanations, good examples and alternate ways of doing things. It covers both software and hardware, to the extent of including (for example) hardware diagrams for video improvement, and for a reset switch.

The support disk contains some of the examples explained in the book, as well as a set of utilities. There's a pretty standard BASIC tool-kit, **Supermon**, a DOS wedge, **Backup 1541**, a disk cataloguer, a **DATA** writer and disk utilities. Some of these are public domain, with acknowledgement of the original authors (good to see). The cataloguer shows start and end addresses — useful for machine language programmers. Otherwise, there's nothing here that isn't in the TPUG library already. The advantage of having the book is the excellent documentation provided for these utilities.

This book is well written, informative and useful. It has a glossary (helpful for beginners), an index, and not too many appendices — that is, it doesn't reproduce the last two hundred pages of the Commodore *Reference Guide*! □

Ollie's Follies
from American Eagle Software
Arcade game for Commodore 64

Review by Michael Quigley

This is a well-designed program that combines elements of 'jumping' games like **Donkey Kong** with the nasty robots of **Impossible Mission**. Your object is to get through various obstacles in twenty-four rooms. Robots aside, hazards such as drainpipes, lightning bolts, lasers, moving steps, elevators, balance beams and fans impede your progress. In order to escape to the next screen, you have to open the exit door (the path to which is fraught with more peril) with a key.

When I first started playing the game, I was intimidated by the prospect of twenty-four screens with no apparent bonuses, so I turned it over to my testing staff. They soon discovered that progressing to the end of Level 4 resulted in a clue to starting the game at the fifth level; and also that logical deduction led to another clue, which made it possible to start at Level 9.

The game has plenty of imaginative touches, as well as a few peculiar ones. In order to defeat the robots, you have to jump up and touch the 'energizer', which is like a giant fright wig. Its effect lasts for a few seconds while you are wearing it.

The ultimate object of the game is a mystery to me. However, for the arcade game junkie, its numerous challenges will provide hours of amusement... and frustration. □

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.

King Chip

King Chip from XYLYX Computer Entertainment Ltd., 20 Torbay Road, Markham, Ontario L3R 1G7. Price: $39.95 (Cdn.)

King Chip comes in a heavy box with a mysterious description on it: "The computer game that doesn't require a computer". This is sufficient to make anyone open the box — revealing 675 cards with questions and answers, six playing boards, forty-two chips, two plastic board connections and one die. As you may have guessed by now, **King Chip** is a trivia-type game, modelled after **Trivial Pursuits**, that tests players' knowledge of computers and high technology. To be more precise, the categories covered in this trivia contest are as follows: Data Communications, History and Current Events, Hardware, Jargon and Acronyms, Potpourri and Software. The question and answer cards are divided into five levels of difficulty.

SG-10C Upgrade Kit

SG-10C Special Edition Kit from Star Micronics, Inc., 200 Park Avenue, Suite 3510, New York, New York 10166-0186. There is no charge for the kit.

Here is some good news for SG-10C printer owners. Star Micronics has recently announced the **SG-10C Special Edition Kit** to expand the versatility of their Commodore-dedicated SG-10C printer.

The kit contains two ROM (Read-Only-Memory) chips to be installed into the printer. They provide the following new features: near letter quality, italics, double-density bit-image graphics, superscripts and subscripts, double strike, variable line spacing, and underlining.

A small eight-page manual gives all the necessary installation instructions, and the escape codes to select the new features.

For general information about the **SG-10C Special Edition Kit**, call the nearest Star regional sales office: New York Sales 212-986-6760, Chicago Sales 312-773-3717, Los Angeles Sales 714-586-8850. For technical data, call Star's technical support headquarters at 714-768-4340.

Kernal Tool Kit

COMPUTE!'S VIC 20 and Commodore 64 Tool Kit: The Kernal, by Dan Heeb, published by COMPUTE! Books, PO 5406, Greensboro, North Carolina 27403. Price: $16.95 (US).

This 429-page book, a companion to an earlier volume covering BASIC, gives a thorough description of the built-in Kernal routines that make up the operating system of the C-64 and VIC 20 computers. The book details Kernal operation in ten chapters covering NMI and IRQ interrupts, system reset and the i/o routines. □

Still Sizzling...

C-128 Software

The C-128 is getting more and more support from software companies.

- Timeworks has announced the release of the upgraded **Word Writer 128**, a word processing system that works in C-128 mode with either a 40 or an 80 column monitor. Additional features include on-screen highlighting, right margin justification, superscripts and subscripts, headers and footers, and speed keys for more experienced users. **Word Writer 128** includes an 85,000-word spelling checker. This word processor can be used by itself, or with Timeworks **Data Manager 128**, **Swiftcalc 128** and **Sylvia Porter's Personal Financial Planner**. The retail price for each of these programs is $69.95 (US).
- Abacus Software has released **BASIC-128** , a compiler that Abacus says will speed up BASIC programs from 5 to 35 times. The program compiles to either 8510 machine code, p-code, or a combination of both. It uses integer and formula optimizing techniques, and is said to be completely compatible with BASIC 7.0. Suggested retail price is $59.95 (US).
- Abacus has also announced the release of the C-128 version of its **Chartpak** program, a package for professional charts and graphs (see Products Received, November 1985). **Chartpak** gives the user three times the resolution of the earlier C-64 version: now the entire chart or graph can be viewed, and the screen can be scrolled to see the higher resolution detail. With the extra memory of the C-128, more data can be entered also. Suggested retail price: $39.95 (US).

Superforth 64

Parsec Research has released **Superforth 64 + Artificial Intelligence (AI)**, an integrated programming environment for C-64 and C-128 users. Superforth, an enhanced version of the standard Forth language, is the core of a package also including AI, Graphics I/O and Extended Math modules.

According to Parsec, the i/o program is engineered to offer complete command of all RS 232 communications, printer/plotter word sets, and KoalaPad graphics. The Extended Math module includes matrices and multidimensional lattices, an algebraic expression evaluator, automatic scientific notation, floating point and trigonometric expressions.

The price of the package is $99.00 (US). It is available from Parsec Research, 41805 Albrae Street, Fremont, California 94538, (800)-633-6335.

For Ham Operators

In the last Products Received section (December 1985) we printed an announcement about Jim Grubbs' *Command Post*. Unfortunately, the address of the publisher was not included. We apologize for this oversight. The address is: QSKY Publishing, P.O. Box 3042 Springfield, Illinois 62708. □

Calendar of TPUG Events

Meeting Places

Brampton Chapter: Brampton Public Library, Four Corners Branch, 65 Queen St., on the second Thursday of the month, at 7:30 pm.

Business Chapter: TPUG Office, 101 Duncan Mill Rd., Suite G-7, Don Mills, on the third Tuesday of the month, at 7:30 pm.

Central Chapter: The Central Chapter will no longer be meeting.

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

Commodore 128 Chapter: TPUG Office, 101 Duncan Mill Rd., Suite G-7, Don Mills, on the first Wednesday of the month, at 7:30 pm.

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.

Communications Chapter: TPUG Office, 101 Duncan Mill Rd., Suite G-7, Don Mills, on the fourth Thursday of the month, at 7:30 pm.

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.

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 third Monday of the month, 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., on the third Thursday of the month, at 7:30 pm.

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. □

FEBRUARY

MON	TUES	WED	THURS
3	4 VIC 20	5 C-128	6
10 Eastside	11 Hardware	12	13 Brampton
17 New Users	18 Business	19 SuperPET	20 Westside
24 Commodore 64	25	26	27 COMAL Communications

MARCH

MON	TUES	WED	THURS
3	4 VIC 20	5 C-128	6
10 Eastside	11 Hardware	12	13 Brampton
17 New Users	18 Business	19 SuperPET	20 Westside
24 Commodore 64	25	26	27 COMAL Communications

Bulletin Board

Microsearch

Microsearch®, a comprehensive database containing more than 20,000 abstracts about personal computers and related products, is now available on the CompuServe Information Service.

The abstracts include product reviews from more than 200 microcomputer-related periodicals, as well as product literature from more than 4,500 manufacturers and software publishers. The information provided refers to product availability and compatibility, software application and product evaluation.

Microsearch is organized into three categories:

- Software Information
- Hardware/Services/Accessories Information
- Directory of Manufacturers

All types of software products, including business and professional applicatons, games and home software are covered in Software Information. Hardware/Services/Accessories Information includes everything from modems to online services. The Directory of Manufacturers provides company names, addresses, contacts' names and toll-free telephone numbers when possible.

Subscribers can access the CompuServe Information Service with a personal computer, terminal, or communicating word processor; communications software; and a modem. Also, in addition to the regular CompuServe connect rates, Microsearch will carry a $10 per hour surcharge.

Kids Computer News

Thomas F. Trocco, who hosts 'Sam's Computer School' on *WNYC-FM Kids America*, announces a new national computer newsletter. Entitled *Kids Computer News*, it features reviews of computer software by *Kids America* listeners, as well as the latest computer news.

Kids Computer News is a monthly publication, and the subscription cost is $5.00 (US) per year. For more information about the newsletter, contact: Thomas F. Trocco, Science Dept. Chair, St. Hilda's & St. Hugh's School, 619 West 114th Street, New York, New York 10025, (212)-666-9871.

Racter

As this issue's feature is Artificial Intelligence, we would like to remind you of **Racter**, whose book of poems, dialogues and limericks was favourably reviewed in our March 1985 issue. If **Racter**'s name doesn't ring a bell, let us tell you that **Racter** is the first computer program to write a book. **Racter** is now available for IBM PC and Apple computers, including the Macintosh.

The number of **Racter** fans has been growing ever since the program appeared on the market. To let them share their experiences with **Racter**, the Institute of Artificial Insanity has been formed. Mindscape, **Racter's** publisher, claims that the Institute's membership is expanding rapidly. Let's hope that a Commodore version (perhaps for the Amiga) will come out soon, so that new inmates may be admitted to the Institute. □

Classifieds

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.

For Sale: Commodore SuperPET, 4040 dual disk drive, 8010 modem, Wordpro 4+, Visicalc, Flexfile database and lots of other software. $2100.00 or best offer. (416)-446-1035.

WANTED: 8032, 8050 or 8250. Call Maria (416)-364-9020.

For Sale: PET 4032 with 80 columns, 4040 disk drive, 4022 printer. Excellent condition. $1200.00 (Cdn.) Call (313)-687-0236.

advertisers' index

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 Contacts

TPUG OFFICE 416/445-4524
TPUG BBS 416/273-6300
TPUG MEETINGS INFO 416/445-9040

Board of Directors		
President	Chris Bennett	c/o 416/445-4524
Vice-President	Gerry Gold	416/225-8760
Vice-President	Carl Epstein	416/492-0222
Recording Sec.	Doris Bradley	416/782-7320
	David Bradley	c/o 416/445-4524
	Richard Bradley	c/o 416/445-4524
	Gary Croft	416/727-8795
	Mike Donegan	416/639-0329
	John Easton	416/251-1511
	Keith Falkner	416/481-0678
	Louise Redgers	416/447-4811
General Manager	Bruce Hampson	416/445-4524
TPUG Magazine		
Publisher	Bruce Hampson	416/445-4524
Editor	Nick Sullivan	416/445-9865
Assistant Editor	Tim Grantham	416/445-9865
Production Manager	Astrid Kumas	416/445-9865
Ad Sales Reps	Hunter Nichols Inc.	416/439-4140
Meeting Co-ordinators		
Brampton Chapter	Jackie Bingley	c/o 416/445-4524
C-64 Chapter	Louise Redgers	416/447-4811
COMAL Chapter	Donald Dalley	416/742-3790
	Victor Gough	416/677-8840
Communications	Darrell Grainger	c/o 416/445-4524
Eastside Chapter	Judith Willans	c/o 416/445-4524
	Nina Nanan	c/o 416/445-4524
Hardware Chapter	Frank Hutchings	c/o 416/445-4524
SuperPET Chapter	Gerry Gold	416/225-8760
VIC 20 Chapter	Anne Gudz	c/o 416/445-4524
Westside Chapter	John Easton	416/251-1511
	Al Farquharson	519/442-7000
Business Chapter	Louise Redgers	416/447-4811
New Users Chapter	Louise Redgers	416/447-4811
C-128 Chapter	Adam Herst	c/o 416/445-4524
Librarians		
COMAL	Victor Gough	416/677-8840
PET	Mike Donegan	416/639-0329
SuperPET	Bill Dutfield	416/224-0642
VIC 20	Richard Best	c/o 416/445-4524
Commodore 64	Derick Campbell	416/492-9518
B-128	Paul Aitchison	c/o 416/445-4524
Amiga	Mike Donegan	416/639-0329
C-128	Adam Herst (CP/M)	c/o 416/445-4524
TPUG Bulletin Board		
Sysop (voice, weekdays)	Sylvia Gallus	c/o416/896-1446
Assistant Sysop	Steve Punter	c/o416/896-1446

TPUG library disk
( )
Toronto PET Users Group Inc.
101 Duncan Mill Road, Suite G7
Don Mills, Ontario, Canada M3B 1Z3
Tel: (416) 445-4524

TPUG LIBRARY DISKS

10 months of the latest, fabulous public domain software only

$79.95 CANADIAN
US orders US$59.95

Subscribe now to start receiving the TPUG disk of the month

You SAVE 20% or more by using this offer as a TPUG member

THIS OFFER VALID FOR PAID UP TPUG MEMBERS ONLY
PLEASE PRINT

Name: ______________________________ Membership # ______________

Address: ______________________________

____

Type of Payment

□ Cheque Amount $ ______________
□ Mastercard Currency □ Can. □ US
□ Visa Credit Card # ______________
□ Money Order Expiry Date ______________

Signature ______________________________

Type of Computer

□ C 64
□ VIC 20
□ PET □ 4040 □ 8050
□ SUPERPET (10 disks) □ 4040 □ 8050
□ C128 (1541 only)
□ MS/DOS
□ AMIGA (3½") (7 disks) □

HOW TO GET OVER $2000 WORTH OF NEW CAPABILITIES FOR YOUR COMMODORE 64™

FOR $599*

The Spartan™ is the Apple™ II+ emulator for your Commodore 64™ that will open up a whole new world of hardware and software for you! Imagine adding these features to your Commodore 64™ for the Spartan™ price of $599: □ Apple™ II+ hardware and software capabilities □ 64K RAM expansion □ four software selectable Commodore 64™ cartridge slots □ non-dedicated 8-bit parallel port □ standard audio cassette deck capabilities for your C-64™. The suggested retail value of comparable products offering only these capabilities is over $2200.00* — but the Spartan™ gives you much, much more! By building on your investment in your Commodore 64™— an excellent introductory computer — you create a whole new system with both C-64™ and Apple™ II+ capabilities. There is a whole other world out there! The huge selection of Apple™ II+ hardware and software is now yours to explore! Call toll free for the Spartan™ dealer nearest you.

MIMIC

FOR INFORMATION WRITE:

MIMIC SYSTEMS INC.
1112 FORT ST., FL. 6L
VICTORIA, B.C.
CANADA V8V 4V2

TO ORDER CALL:

1-800-MODULAR
(663-8527)

*All prices quoted are in U.S. funds, freight and taxes not included. Value of components equivalent to the Spartan™ system are quoted from Apple™ II+ CPU and Apple™ II+ single disk drive 1983 list prices, and from current suggested list prices and component specifications of other peripheral manufacturers. Commodore 64™ and Commodore logo are trademarks of Commodore Electronics Ltd. and/or Commodore Business Machines, Inc. Apple™ II+ is a trademark of Apple Computer Inc. Spartan™ is a trademark of Mimic Systems Inc. and has no association with Commodore Electronics or Apple Computer Inc. The Spartan™ is manufactured by Mimic Systems Inc. under license granted by ATG Electronics Inc. of Victoria, B.C. Canada.