

COMAL TODAY

26



Think About It!

COMAL Today
5501 Groveland Ter
Madison, WI 53716

Bulk Rate
U.S. Postage
Paid
Madison, WI
Permit 2981

If the label says
Last Issue 26
You must renew now.
Use the order form inside.
No other notice is sent.



page 47

COMAL TYPE DETECTOR

page 33

C64 PRINTER TIPS

page 12

CONVERTER

page 37

AmigaCOMAL 2.04

page 18

CHALLENGE

page 47

PEOPLE LINK

page 50

PARAMETERS

page 10

COMPATIBILITY

page 32

AMIGA FISH DISKS

page 17

IBM PC COMAL 3.02

page 22

COMAL Support Club - \$25 Free Time!

Instant access to new COMAL programs. Answers to your questions quickly. Three monthly meetings with other COMAL users. This is possible with the PLink national online network. All you need is a modem on your computer. We have arranged a full \$25 free time for every *COMAL Today* subscriber (see ad below). We even have a good terminal program on *Today Disk 26* for IBM (specify disk type), C64 or Amiga.

Amiga, C64, IBM COMAL programs are available. Programs from *COMAL Today* are available even before you get the issue! PLink has clubs for each computer. Amiga Zone is famous (see ad below).

Workshop - help with your programs! This is where the Converter program in this issue was developed. A small group of people can simultaneously work on a program using the **workshop** as a common holding area for the latest revision of the program.

Questions? Post a question. Check back the next day to find one or more replies. Help from many sources ... not just from the COMAL CLUB staff! Write your

question with your word processor at your leisure. Save it as a standard text file. Then have your terminal program ASCII send it to the Notice area on PLink... quick and easy.

National Conferences - Come to the national COMAL meetings every 1st Wednesday and every 3rd Thursday at 9pm Central time. Or have spontaneous conferences with other users who are signed on at the same time as you. The /LIST command shows who is using the COMAL CLUB.

Real Time FUN Challenge - Put the fun back into programming. Every first night of each month we sponsor a CHALLENGE. If you are an advanced programmer this is your chance to solve the problem faster than the COMAL CLUB staff. Beginners will have a chance to learn about programming since everyone is working on the same problem.

Prepare. Read the articles in this issue about PLink. Then try it with \$25 free online time! Call the number listed in the ad below and mention COMAL.

You've seen the rest... now try the BEST!

A M E R I C A N P E O P L E

SPECIAL OFFER FOR COMAL MEMBERS

Sign up NOW and receive \$25.00 of online credit! Just call our modem signup number (below) and when asked "Where did you hear about us?" just type in COMAL. Your account will automatically be credited with \$25.00 of FREE online connect time.

"THE KING OF THE HILL"

For the Amiga, PLink is by far the largest and most active network.... Its public domain library is one of the best around with thousands of files ready for downloading. Of the four networks discussed... PLink has the lowest signup fee and some of the lowest hourly rates. Plink offers a lot for your money.

- Lou Wallace, *AmigaWorld*, 5/89



People/Link's AmigaZone **NOTICE BASES** are a hotbed of information. Need Help? Simply post your question, and chances are your problem will be solved by the next time you log in. Talk directly to industry leaders. Software authors, hardware engineers, and technical and support reps hang out in the AmigaZone! People from dozens of Amiga products companies like *Gold Disk*, *Lattice*, *WordPerfect*, *ASDG*, *MSS*, *NewTek*, *PP&S*, *Microillusions*, and *New Horizons*. Writers and editors from *AmigaWorld*, *INFO*, *Amiga Resource*, *Computer Shopper*, *Amazing Computing*, *Jumpdisk*, and others use the AmigaZone as a prime source of information. You can, too.



If it's files you're after, you'll think you've gone to heaven. The AmigaZone **LIBRARIES** set the standard - the newest programs, artwork, demos, animations, and utilities of all descriptions await your downloading pleasure. Our fast Windowed Xmodem transfer protocol is now part of all the best and most popular terminal software, and it is **FAST!** All AmigaZone files are sysop-tested for your peace of mind. Care to share your own creations? Uploading is **FREE** during non-prime connect time. Unlike a BBS there's no required "ratio" of uploads to downloads. Feel free to grab all the hottest freely-distributable software you've read about.



People/Link is PEOPLE oriented. The AmigaZone's **LIVE CONFERENCES** are where you can meet your friends or make new ones, live, online. Every evening is devoted to a special topic like Music & MIDI, Programming, Graphics & Video, Games. And every Sunday night it's "AmigaMANIA" - dozens of fellow Amigans online with you from all over the world. Chat with those famous folks you've always wanted to meet! Bill Hawes helps ARexx users in a special class each month. It's a cozy atmosphere with lots of friendly people who can help you get the most out of your Amiga experience.

Sign up NOW or for more info, call People/Link: **BY VOICE:** (800) 524-0100 or (312) 648-0660, 9:00am to 5:00pm, Mon-Fri, Central Time. **BY MODEM:** (800) 826-8855, 24 hours a day at 300/1200/2400 baud.

PEOPLE/LINK is a service mark (sm) of American Home Network.



ISSUE 26

General

- 2-5 - Editor's Disk - Len Lindsay
- 6-9 - Letters
- 10 - Parameters - David Warman
- 32 - Compatibility - David Warman
- 33 - COMAL Type Detector
- 37 - Converter - David Warman
- 47-49 - Challenge
- 61-64 - Order Form

On-Line Support

- 0 - People Link Special Offer
- 7 - Terminal Scripts - David Warman
- 50-52 - People Link Overview - David Warman
- 53 - People Link Tips - Mitch Pilchuk
- 54-56 - People Link Walk Through - Mitch Pilchuk
- 57-60 - Redi-Access and TeleNet Phone Numbers List

Amiga

- 6 - AmigaCOMAL Packages - Jim Boyd
- 7 - AmigaCOMAL 2.04 File Note - David Warman
- 8 - Packages FROM Warning - David Warman
- 9 - Public Domain Programs/Comal - Jack Baldrige
- 17 - Amiga Fish Disks - A Goldmine of Programs
- 18-20 - AmigaCOMAL 2.04 - Some of the Changes
- 21 - AmigaCOMAL 2.04 - Svend Pedersen
- 33 - COMAL Type Detector
- 37 - Converter - David Warman
- 65-66 - Fish Disk Reference Chart - Jesse Knight

IBM

- 7 - UniComal 3.0 Static Scope - Nicholas Seachord
- 8 - UniComal 3.02 HEAPSIZE - UniComal
- 22-24 - UniComal
- 25 - UniComal 3.02
- 26 - UniComal 3.02 Chart
- 27-29 - Getting Used to UniComal 3.02 - Tom Kuiper
- 30-31 - UniComal 3.0 Notes - Craig Van De Grift
- 33 - COMAL Type Detector
- 37 - Converter - David Warman

C64

- 6 - COMAL and the 1581 - Paul Baker
- 12-16 - C64 Printer Tips - Rea, Warman, Lindsay
- 33 - COMAL Type Detector
- 37 - Converter - David Warman

COMAL TODAY

<u>Editor</u>	<u>Contributors</u>	<u>Contributors</u>
Len Lindsay	Paul Baker	Mitch Pilchuk
	Jack Baldrige	Joel Rea
<u>Assistant</u>	Jim Boyd	Nicholas Seachord
Maria Lindsay	Jesse Knight	Geoffrey Turney
	Tom Kuiper	UniComal
<u>Artwork</u>	Len Lindsay	Craig Van De Grift
Rhianon Lindsay	Svend Pedersen	David Warman

COMAL Today is published by COMAL Users Group, U.S.A., Limited, 5501 Groveland Ter, Madison, WI 53716 and welcomes contributions of articles, manuscripts and programs which would be of interest to readers. All manuscripts and articles sent to COMAL Today will be treated as unconditionally assigned for publication and copyright purposes to COMAL Users Group, U.S.A., Limited and is subject to the Editor's unrestricted right to edit and to comment editorially. Programs developed and submitted by authors remain their property, with the exception that COMAL Users Group, U.S.A., Limited reserves the right to reprint the materials, based on that published in COMAL Today, in future publications. There will be no remuneration for any contributed manuscripts, articles or programs. These terms may be varied only upon the prior written agreement of the Editor and COMAL Users Group, U.S.A., Limited. Interested authors should contact the Editor for further information. All articles and programs should be sent to COMAL Users Group, U.S.A., Limited, 5501 Groveland Ter, Madison, WI 53716. Authors of articles, manuscripts and programs warrant that all materials submitted are original materials with full ownership rights resident in said authors. No portion of this magazine may be reproduced in any form without written permission from the publisher. Local Users Groups may reprint material from this issue if credit is given to COMAL Today and the author. Entire contents copyright © 1990 COMAL Users Group, U.S.A., Limited. The opinions expressed in contributed articles are not necessarily those of COMAL Users Group, U.S.A., Limited. Although accuracy is a major objective, COMAL Users Group, U.S.A., Limited cannot assume liability for article or program errors.

Please note these trademarks: Commodore 64, CBM of Commodore Electronics Ltd; Amiga of Commodore Business Machines, Inc; Calvin the COMAL Turtle, Captain COMAL, Super Chip, COMAL Today, Doc Box, Common COMAL, Power Driver, NET-2-NET, Fish To Go of COMAL Users Group, U.S.A., Limited; CP/M of Digital Research; IBM of International Business Machines; Apple, MacIntosh of Apple Computer Inc; QLink, Quantum Link of Quantum Computer Service; PLInk of People Link; CompuServe of Compuserve; Prodigy of Prodigy; GEnie of General Electric; Word Perfect of Word Perfect Corp; UniComal of UniComal; LaserJet of Hewlett Packard; Sorry if we missed any others.

Our NEW Address is:

COMAL Users Group, U.S.A., Ltd.
5501 Groveland Terrace
Madison, WI 53716
(608) 222-4432

Editor's Disk

by Len Lindsay

You have the latest issue of *COMAL Today* in your hands. There were delays in getting this issue out. Without a staff, I have to manage everything myself. Keeping the user group going is a big job. Editing new books is, too. So is editing and publishing a newsletter. Toss in order processing, packing and shipping. Plus signing onto our on-line COMAL Support Network (PLink) daily. It keeps me busy.

On Vacation Dec 21, 1990 - January 6, 1991

I'm on vacation for a couple of weeks during the holidays. Since I am the only one here, that means the phone won't be answered, and mail will pile up.

Instant On-Line COMAL Support is now available on PLink.

COMAL Support Club

Hopefully our online support will make up for any delays in the printed copy of *COMAL Today*. Programs in this issue were available in the People Link (PLink) COMAL CLUB before this issue was even printed! Many of the articles were available to read online early as well! Over 100,000 characters of questions and answers per month pass through the Notice Area. I check in daily to answer questions, as do many other COMAL users.

Articles and programs being prepared for next issue will show up on PLink as I work on them. If you want advance COMAL information, PLink is the place to be.

Which Net?

We checked out other networks before deciding on PLink for our support center.

☛ Quantum Link (QLink) is only for C64/128 computers. We still maintain the largest COMAL program library there! However, it excludes Amiga and IBM users. On-line meetings are every first Sunday and second Thursday at 9pm Central.

☛ Prodigy is cute. However it is extremely SLOW. You can't prepare your messages offline on your word processor or print off things as you read them. Nor can you store the text as a disk file. Plus, it would be messages only, no library of program files! Worse yet, virtually every screen includes advertising! Besides, it is Mac and IBM only! ☺

☛ CompuServe is too expensive and business oriented. We inquired about setting up an official COMAL section there, but never received an answer. There are COMAL users on CompuServe however, especially in the Amiga area.

☛ We are trying to get a COMAL area on GEnie, but so far have had no luck. Meanwhile users have advised us that PLink is easier to use and is just as affordable.

People Link (PLink) - Affordable - Even FREE

PLink has been very helpful. They gave us 4 meg of storage for online COMAL programs. In addition to that they provided us with a Notice Area for announcements plus questions and answers. They also gave us our own private Conference Area for meetings. You can use that area for your own informal meetings any time you wish (yes, *PLink is accessible 24 hours per day, though prime time rates are high*).

Get \$25 of FREE On-Line time on PLink, just say COMAL sent you.

☛ Best of all, PLink has offered every *COMAL Today* subscriber \$25 of free online time just to try out our COMAL Support Club. Of course you can use any of the PLink services, such as Travel and Shopping services. Amiga users are fortunate that PLink is famous for being first with new Amiga programs: /GO AMI. You also may be interested in the Science Fiction & Fantasy club: /GO SF.

The authors of many programs are available on PLink. For example, TARTAN wrote **COM-AND** (*the IBM term program*) and BAUDMAN wrote **Baud Bandit** (*the best Amiga term program*).

PLink Tips

When you sign up, make sure to say COMAL sent you to get your \$25 free on-line time.

Remember it costs more to use PLink at 2400 baud than at 1200 or 300. Connect at 300 baud while you are learning how to get around and save money. Then use 2400 baud to do uploads and downloads quickly. Use 300 baud when attending our on-line national meetings!

Use 300 baud at on-line conferences.

The first time you sign on to PLink, make sure you issue the command: /GO 411 ... this takes you to a special section that lets you assign your PLink account with an ID NAME of your choice (up to 10 characters). Much better than the cryptic WSE128 type ID you are assigned by their computer.

The first time on PLink, type this command: /GO 411

Post your questions about PLink in our COMAL CLUB and get quick answers!

Instant AmigaCOMAL Updates

We set up a special section in the COMAL CLUB on PLink just for updates. Only registered AmigaCOMAL users are allowed into the library where they can download the latest updates to AmigaCOMAL for free (only costs the connect time while downloading). We may make this feature available for IBM COMAL users as well. You save the \$10 cost of the update, plus you get it immediately!

Everyone is allowed in the UPDATES NOTICE area. Here we will post announcements about the latest updates, plus you and other COMAL users can post comments about these updates.

**FREE update to AmigaCOMAL.
Pay only the normal connect time charges.**

Free AmigaCOMAL Compiler Offer

For a special limited time, we are including the compiler and Developers set with every standard AmigaCOMAL system. It is usually a \$35 option but is included for no extra charge now. If you purchased AmigaCOMAL without the compiler option earlier ... we have a special offer on the compiler for you, too ... either FREE or \$9.95. You can download the compiler upgrade from our PLink UPGRADE section (only pay the connect time).

FREE AmigaCOMAL Compiler with every AmigaCOMAL system.

Or for \$9.95 we will send you the compiler disk with its small manual. If you order this, you must be a registered AmigaCOMAL user and must clearly state you wish to get the compiler disk and manual special offer for \$9.95 (which includes the shipping and handling charges).

Intro To COMAL

The 2nd edition has been nearly done for about a year. All I have to do is a final proof pass of the text and then lay out each of the 350 pages. Don't be deceived though. That is an enormous task. I hope to get it to the printers after this goes to press. The book is excellent, and now applies to ALL versions of COMAL. Such a book is sorely needed. If I only had a staff, I could

COMAL Today INDEX - part 2

David Warman is finishing up work on part 2 of the index to COMAL TODAY ... covering issues 13 through 26. The original index is still available in limited quantities (for issues 1-12).

Amiga

★ To help finance our COMAL activities we have become the premier source for Fish Disks. We have the entire set available for just \$1.95 each, no minimum, no shipping fee for UPS ground (25¢ per disk for First Class). You probably get Fish Disks now and then (*if you don't, you're missing great programs*). Support COMAL by getting them from us. No one beats our low price! Jesse Knight created a Super Fish Reference List using AmigaCOMAL! It is featured on our back cover.

Amiga Programs - A Goldmine of them - 400 disks full already in the Fish Disk collection.

If you are unfamiliar with Fish Disks, see the article on them in this issue ... then make sure you order Disk#301 (with Aquarium) and ask for the special offer of a free Fish Catalog Disk! While you are at it, get a couple more Fish Disks to get a feel for this popular library of Amiga programs. When you order 4 or more disks, we ship them in a special sized box that is great for storing your disks (holds 40 disks).

■ You can subscribe to the Fish Disk collection too! That way, whenever new disks are released you will get them right away! Just send us a letter stating that you wish to place a standing order for Fish Disks. You must provide us with a VISA or MC charge number and expiration date. You will be charged for the disks as they are released and shipped to you. You must advise us immediately if you change credit card accounts or if you get a new expiration date (every year or two). This is a great way to keep up to date on new Amiga programs for a very minimal cost.

Subscribe to Fish Disks now! Keep up to date automatically.

★ Best Selling Fish Disks ★

We've been selling Fish Disks for some time now. The following are the 13 most requested disks:

- 262 - Indent, Jumble, Lotto, QMouse, QView, World Data Bank
- 269 - Calendar, ChessTutor, PropGadget, RadBoogie, ShowDisk, SpinPointer
- 274 - HP11, KeyBiz, LookFor, SlideMaster, Snap, Xoper (Snap update is on disk#326)
- 288 - DiskSpeed, PlotData2D, Scriptit
- 289 - AmiGo, Atree, LHarc, Orbit, TreeWalk (LHarc update is on disk#383)
- 295 - GnuGrep, LHwarp, MandelMountains
- 303 - Mandelbrot, CPM, DEM, Demon, FixIcons, IceFrac, Rocket, ScreenZap, SnowFall
- 336 - Car, FileWindow, MiniBlast, Sys
- 337 - C Manual
- 349 - MED (Music Editor)
- 350 - Icons, MemMometer, Stichery, TrackUtils
- 354 - FastBlit, MandelMountains, MemGuard, MXMLib
- 355 - AlgoRhythms, NComm

Today Disks

Every *Today Disk* (starting with #25) is available for IBM (*specify disk type*), Amiga and C64. *Today Disk* 26 includes a modem terminal program (*the favorites of the PLink COMAL Club Staff*) that you can use to access PLink. Tentatively they are:

IBM PC: COM-AND (better than ProComm)

Amiga: NComm (complete with scripts)

C64: CCGMS, WXModem, or M&MTerm

(I think at least two of them will fit on a disk)

Amiga note: If you want the **best** modem program, get **Baud Bandit** (v1.50). To use it properly you need **ARexx**. Baud Bandit is the term program that I use (*and so does Mitch and David, fellow PLink COMAL Club staffers, as well as the staff of the Amiga Zone*). If you plan to reply to notices or to take part in on-line meetings, Baud Bandit is super! It can even do batch downloading and automate reading notices.

News

UniComal expanded into an office complex (*well, actually they took over a multi story house in*

Continued...

Editor's Disk

...Continued

Copenhagen) and now has about double the staff as last year. They even have a FAX, so you can FAX comments to them! (Just call 011-45-31-676314).

UniComal has expanded. You can FAX your comments to them.

Freddy (*the programmer who wrote CP/M COMAL*) has joined UniCOMAL. So has Svend (*who wrote AmigaCOMAL*). UniComal also has good ties with both IBM and Commodore. That is great. However, IBM pricing seems to have too much influence on them! Their COMAL for IBM PC is targeted at businesses ... and priced like it. I have to admit that it is the best COMAL out now (*the latest release, 3.02, even includes menus and windows as options!*)

Latest COMAL Updates:

IBM PC COMAL 3.02 AmigaCOMAL 2.04

IBM PC COMAL has been updated to version 3.02. AmigaCOMAL has been updated to version 2.04. The AmigaCOMAL update is available on PLink to registered AmigaCOMAL users.

New Packages: To get a quick reference list of all the new commands in any package just type this:

IBM type: INTERFACE packname TO "lpt1:"
Amiga type: LISTPACK packname TO "filename"
(leave off the TO xxx part and it goes to the screen)

Future COMAL Today

I use PLink to help get future COMAL Today articles and programs ready. If you can spend some time polishing up programs, and maybe making minor changes so they will work with COMAL on IBM, Amiga and C64, let me know. Just send EMail to Captain C. Arrangements can be made then.

I have a box full of new COMAL programs and articles ... enough for another issue of *COMAL Today* (with the emphasis back on COMAL programs themselves). Watch PLink's COMAL Club for advance access to the programs and articles.

PLink Emphasis

We emphasized PLink in this issue for a good reason. We think it is the wave of the future. You get instant access to the latest COMAL programs and information. You get to conference with other COMAL users, many with *COMAL Today* articles and programs to their credit! (*Plus you can use the many other services that PLink provides.*)

We spend a substantial amount of time maintaining our COMAL Club on PLink. It's a natural extension to *COMAL Today*. We want to make getting started on PLink as easy as possible for you. In this regard we arranged for \$25 of free on-line time. While you are getting used to how the system operates, you don't have to worry about being charged as you make mistakes! We put together articles with tips on using PLink. We even printed the latest list of TeleNet and Redi-Access phone numbers! Check if a number on the list is a local phone call for you. If so, there will be no long distance phone charge for you to access PLink, only connect time.

We check the COMAL CLUB on PLink virtually every day. You get your questions answered quickly, plus you can read answers to other users' questions! You can even order COMAL items and Fish Disks via EMail to Captain C. **You can SEND EMail while you are in the COMAL Club! Just type /MAIL. To send a on-line message type: /MSG.**

All you need is a modem.

All you need is a modem and your computer to try PLink. Terminal programs are on Today Disk 26 and even some scripts to help automate things. The free \$25 on-line time lets you try it without charge. The more that join, the better it will be for everyone. We hope to see you on-line soon!

Letters

COMAL and the 1581 Drive

From Paul L. Baker:

I just purchased a 1581 disk drive for my COMAL 2.00/Jiffy Dos enhanced C64. With JiffyDos, to access the sub-directories, one has only to use the command @/"subdirname" and @/ to return to the root. With COMAL 2.00, the command is:

PASS "/subdirname"

and to return to the root:

PASS "/"

COMAL 2.0 can change subdirs with Jiffy Dos via:

PASS "/subdirname"

PASS "/"

I partitioned a disk for TODAY 8 & 9 and subdivided those partitions for Side 1 and Side 2. "This is slick!", I thought, and tried to CHAIN "hi" from the 2.00 subdirectory. Oops. CHAIN somehow signals the drive to return to the root directory and COMAL reports a "File Not Found" error. I tried the following (which DOES work):

**PASS "/TODAY 8"
PASS "/2.00"
LOAD "HI"
RUN**

Just CHAIN doesn't work. I'm not too upset by this, since the 1581 drive didn't exist when the COMAL 2.00 cartridge was produced, but I can't seem to come up with a fix on my own, yet.

[CHAIN actually re-initializes the disk, thus removing any subdirectory settings you make previously. As you noted, COMAL 2.0 was released before there was such a thing as a subdirectory.]

AmigaCOMAL: Creating New Packages

Plink EMail from Jim Boyd:

I got the COMAL system Thursday (Nov. 14 or so). Unfortunately, I haven't had time to do much with it, YET. I've read parts of the manual and scanned the rest. It doesn't sound like creating new packages for AmigaCOMAL will be too hard.

I'm thinking of adapting the programs from the Public Domain C Manual thing (Fish Disk #337). It is a really neat disk-based 'C' tutorial thingy that shows how to do almost everything: windows, requesters, etc, in 'C'. These programs could be ported over to COMAL. Either that or one could intuitionize some of the programs found in *COMAL Today*. What do you think?

What I find hard to believe is that nobody has done this so far. There are examples in the COMAL manual and everything!

What I'm finding hard to believe is that nobody has done this so far. There are examples in the COMAL manual and everything! Oh well, I shan't look a gift horse in the mouth.

You bet, Jim! We also are surprised that we have not been flooded with nifty packages for AmigaCOMAL yet. It should snowball if we could get it started.

That CManual disk would be great for people to look into ... as well as the many Libraries that are available on Fish Disks. For example, Fish Disk #393 includes: ILBMLib, FuncLib and RexxLib.

*Has anyone out there created their own packages they would like to share? You can upload them to our COMAL CLUB on Plink for ***FREE*** since all upload time is not charged for by Plink. **Yes, absolutely free!** No charge for any connect time involved in UPLOADing files to the COMAL CLUB! Or send a disk to COMAL Users Group.*

Scripts & First Time PLink Signon

PLink EMail from David Warman:

About the logon scripts that will be included on *Today Disk #26*. Scripts require a fixed environment, that is, PLink must always supply the same prompts in order for the script to do its thing. You should warn people that they may not be able to use the scripts the first time they sign onto PLink or the first time they visit the COMAL Club.

The first time you sign onto PLink, some scripts/macros may not work. Some also may require /ADV mode.

I think there is some special text that you are presented with the first time you call PLink. It's possible that the script may still work correctly if it just waits for the **MAIN MENU>** prompt. It wouldn't make any difference how much extra stuff you had to read through in that case.

The first time you log into the COMAL Club you are presented with the welcome information and asked if you want to join. The script may still work, as long as the **CLUBS>** prompt isn't presented until all the preliminary stuff was out of the way. The scripts might work on the first call to PLink, but it's safer to advise against using them the first time.

AmigaCOMAL 2.04 Update file note

PLink EMail from David Warman:

There have been some changes in the internal tokenization of programs in Version 2.04. As a result you cannot LOAD programs that were SAVED in the earlier version (2.0+). You must first run AmigaCOMAL 2.0. Then LOAD your SAVED programs and LIST them to disk. Later you can run AmigaCOMAL 2.04 and ENTER the programs. The programs then should be fine and you can SAVE them from within 2.04. [You may wish to keep a copy of 2.0+ handy to use on 2.0 programs on PLink.]

An AmigaCOMAL 2.0+ program must be LISTed to disk and ENTERed into the new 2.04 Update COMAL system. Likewise for COMAL coded packages.

The same thing goes for packages written in COMAL. The packages in version 2.04 are OK, but before you can **USE** any packages you may have written with version 2.0, you must **LOAD** them into version 2.0 AmigaCOMAL and **LIST** them to disk, then **ENTER** them into version 2.04 where you can then **SAVE** them. But this time, save them with a **.CMP** extension instead of **.PCK** since the **.PCK** extension now signifies a machine language package. **.CMP** signifies a package written in COMAL. [Remember that **LISTPACK** lists the header line of **PROC** and **FUNCs** you define in your COMAL coded packages, so use readable parameter names! Try it. Type: **USE packname** then **LISTPACK packname**.]

AmigaCOMAL 2.04 now has two types of package filename extensions. **.CMP signifies a package written in COMAL. **.PCK** signifies a ML package.**

IBM PC COMAL 3.02 and Static Scope

Mail from Nicholas Seachord:

I wrote a database program in version 2.2 and decided to ENTER it into 3.02. It loaded with no problems, but there were a few new bugs in it. After hours of debugging, I discovered the problem. My array pointers were pointing to the wrong places! The program below illustrates this problem. A parameter must be used with a FOR loop variable when calling a procedure which lies outside of the loop. The revised code works in both versions of COMAL. [Note that it is an alias type parameter.]

```
FOR x=1 TO 5
  showit
ENDFOR x
//
PROC showit
  PRINT x;
ENDPROC showit
```

===== Correct ... with parameter =====

```
FOR x=1 TO 5
  showit(x)
ENDFOR x
//
PROC showit(num)
  PRINT num;
ENDFOR showit
```

Yes ... with Static Scope the 3.0 versions of COMAL require you to be more precise with your code. When using a FOR loop variable in outside procedures, make sure you pass the value as a parameter.

With Static Scope, make sure you pass a FOR loop variable as a parameter if you wish to use it in any PROC or FUNC called from inside the FOR loop.

AmigaCOMAL packages FROM warning

PLink EMail from David Warman:

I have more information on the AmigaCOMAL bug I mentioned at the conference.

If a program CHAINS to another program that USES a package FROM a specified directory, that directory must be available or you will get the "Software Error -- Task held. Finish ALL disk activity. Select CANCEL to reset/debug" requestor. (I think I have the message memorized.) Of course, selecting CANCEL causes a GURU and a forced reboot.

Version 2.2 yields:

1 2 3 4 5

Version 3.02 yields:

0 0 0 0 0

If the USE statement in the program being CHAINED doesn't have the FROM part, then a requestor will pop up asking for the appropriate disk. If you insert that disk, everything is OK, but if you put the wrong disk in or click on CANCEL you get the GURU-spawning Software Error message again.

Recapping...

If the program being CHAINED to has the line:

USE package FROM "packages;"

the specified directory must be available or else the GURU error results. If the program being CHAINED has a line:

USE package

If the package is not found, a requestor asks you to insert the proper disk. If the wrong disk is inserted or if you click CANCEL, you will get the Software Error GURU, otherwise it is OK.

If you CHAIN to the program from immediate mode, there is no system error. The program just stops and COMAL reports a "file not found" error.

IBM PC COMAL 3.02 HEAPSIZE

Mail from UniComal:

We found that the heap space is set too low in IBM PC COMAL 3.02. This will be corrected in the next version. The thing to do for now is to increase HEAPSIZE in the profile.

You must increase the HEAPSIZE in the profile or programs using MENUS and WINDOWS packages won't work.

Continued...

Letters

...Continued

Amiga Public Domain Programs & COMAL

Mail from Jack Baldrige

I'm enjoying being a newcomer to AmigaCOMAL. If you have an Amiga, you certainly found out quickly after getting it that Amiga programs use disk space at an astonishing rate. Another thing, which you may not yet know, is that a lot of good software is available, in the public domain or as Shareware. Probably the best source for this software is the Fred Fish library (Fish Disks), which at last report contained 400 disks.

Probably the best source for this software is the Fish Disk collection with 400 disks!

One thing this software can do for you is to make programs smaller without ill effects on their performance. One of the first things I did after getting the AmigaCOMAL disk and running a few of the demo programs was to crunch the AmigaCOMAL program. I'm not talking here about using archiving programs like LHarc, which are used to make compressed files for archival purposes. A crunching program, like PowerPacker, compresses an executable program, which can be saved. When you execute that file, it is automatically expanded into memory and run.

Save disk space! PowerPack your executable programs with PowerPacker from Fish Disk 253.

If you list AmigaCOMAL on your system disk, you'll see that it is 95,068 bytes or 189 blocks long. By using PowerPacker, I obtained a program only 131 blocks long or 62,600 bytes -- a savings of about 35%. (This compares to a file size of 59,272 bytes when compressed with LHarc, but the program then is no longer executable.)

Some programs cannot be crunched or may crash your computer if crunched. However, I've been using the crunched AmigaCOMAL program for several weeks with no problems.

Another thing you might like to do is to customize the AmigaCOMAL function keys. A shareware disk editor named NewZap can be used to do this. First load a copy of the AmigaCOMAL program and use the search function to find the place where the function keys are defined (about the 141st sector of the program). Then use the editing function to change the definitions to those you want. Finally, save your changes before you exit the program.

This shows just two ways to use freely distributable software with AmigaCOMAL. If you have other suggestions, we'd like to hear them. The programs NewZap and PowerPacker are available, by the way, on Fish Disks #164 and #253, respectively.

Make changes only to a copy of your program.

Note: never make changes to your original AmigaCOMAL program; only to a copy of it!

[Another method is to use the KeyMapEd program (on Fish Disk #388) to change any of the keys on the keyboard. There is something like 8 combinations per key, as you can define any of the keys with any combination of the CTRL, ALT and SHIFT keys. The problem if you defined too many keys would be remembering which key does what.]

FISH DISKS are available for just \$1.95 each from:

**COMAL Users Group, U.S.A., Ltd.
5501 Groveland Terrace
Madison, WI 53716
(608) 222-4432**

Parameters

by David Warman

Parameters were new to me when I first started learning COMAL. At first they can be a little confusing, but once you get used to them they are very useful. Parameters can add a lot of versatility to PROCedures and FUNCtions. Let's use a simple example:

```
PROC pause
  FOR x:=1 TO 1650 DO NULL
ENDPROC pause
```

This PROC is a very simple delay loop and does not have a parameter. To execute it you would just put the word **pause** on a program line. A disadvantage of this PROC is that it will always pause for the same amount of time.

There may be times when you want a variable delay -- you might want to pause for 2 seconds at one point in a program and for 5 seconds at another. By using a parameter, you can vary the delay period without having to use a different PROC each time. If you've ever programmed in BASIC you are familiar with parameters, although you may call them "arguments". For example, the "15" in "TAB(15)" is a parameter of the built-in TAB command, which causes the PRINT statement to print spaces up to column 15. In COMAL, you can also add parameters to your own user-defined PROCs and FUNCs.

```
PROC pause(seconds)
  FOR x:=1 TO seconds*1650 DO NULL
ENDPROC pause
```

Now if you type **pause(2)** on a program line you will get a two-second delay. **Pause(5)** will give you a five-second pause. By adding a parameter you made the PROC much more flexible, and it doesn't need to be changed for different pause durations. This is only a simple example. There are more accurate ways to pause for a given amount of time (ie, the TIME or TIMER functions).

There are two types of parameters: ALIAS and REFERENCE

Parameters can be of either "alias" or "reference" type. Most parameters are aliases, which means just what it sounds like: they are not the real variable. That may require some explanation. Let's say you want to count the number of spaces in a string. A FUNCtion to do that might look something like the following (remember that

the difference between a PROC and a FUNC is that a FUNC must RETURN some value -- like the built-in SPC\$(x) does). First, type **AUTO** to provide line numbering then type in the following lines. Of course, you don't need to type the caps or indentation, and the line numbers are provided for you:

```
0010 FUNC count'spaces(string$)
0020   spaces:=0
0030   WHILE " " IN string$ DO
0040     spaces:=+1
0050     string$(" " IN string$):= "."
0060   ENDWHILE
0070   RETURN spaces
0080 ENDFUNC count'spaces
```

This FUNCtion counts the number of spaces in the string "string\$" by using the IN function. First, the space count is set to zero in line 20. The expression:

" " IN string\$

returns the position of the first space in the string. As long as there is a space character in the string, the WHILE loop will be executed. If there are no spaces, it won't be executed at all. If the string is "Four score and seven" then the result of the IN operation will be 5, since the first space is the fifth byte of the string. Since a space was found, the space counter is incremented by 1 in line 40. Now, we don't want to count the same space again on the next time through the loop, so it must be changed to something else. Line 50 does that. Since "" IN string\$" evaluates to 5, the fifth byte is changed to a period, and the string is now "Four.score and seven".

The next time through the loop the IN function will return 11 and spaces will be incremented to 2. This is repeated until there are no spaces left in string\$, at which time spaces will equal 4 and the WHILE loop will be exited. Line 70 returns the number 4 to the main program. Here's an example of how to use it.

First, type in the above FUNC, then type SCAN or RUN to add count'spaces to COMAL's name table. Then type these lines in immediate mode:

```
DIM string$ of 20 // only necessary in Power Driver
string$="Four score and seven"
print count'spaces(string$)
4    <-- this is the computer's response
```

In the process of counting the spaces, the FUNC count'spaces changed all the spaces in string\$ to periods, right? Let's see. Type:

Parameters

```
print string$
```

The periods are gone and the spaces are back! Magic?! No, the spaces are still in string\$ because they were **never changed**. The string\$ that you typed in and the string\$ inside the FUNC are **NOT** the same variable. That is what is meant by alias. This may seem confusing, but it has tremendous advantages. In the above example, you probably would not want to destroy string\$ by changing the spaces to periods because you would only have to change them back to spaces if you needed to refer to the original string again. With alias parameters you can do whatever you like to the variable inside the PROC or FUNC and its value in the main program will not change. To better illustrate this, let's change the variable name. Instead of typing string\$ in the above 3 lines, try using test\$ instead:

```
DIM test$ OF 20 // only need with Power Driver
test$="Four score and seven"
print count'spaces(test$)
4    <-- this is the same response!
```

When the string string\$ or test\$ that you typed above gets passed to FUNC count'spaces, either in immediate mode like you just did, or within a program, its value is assigned to the parameter string\$ in the FUNC header in line 10. A brand new variable is created and used as an alias for the variable in the calling statement. Even if they have the same name they are treated as two different variables. Once the FUNC ends, the alias variable ceases to exist and the memory that it used is freed. (Note: COMAL uses two internal stacks for its temporary and long-term variables, so there is absolutely no garbage collection.)

However, there are times when you **do** want the parameter variable to be changed inside a PROC and to stay changed when the PROC/FUNC is finished. For example:

```
PROC lowercase(string$)
FOR i:=1 TO LEN(string$) DO
  byte: :string$(i)
  IF byte > =97 AND byte < =122 THEN
    byte:-32
    string$(i):=CHR$(byte)
  ENDIF
ENDFOR
ENDPROC
```

This PROCedure changes capital letters (in PetASCII ... for true ASCII, the range is 65-90 and you add 32 to byte) to lowercase letters. Naturally when you call this PROC you want the letters to stay changed, not pop back to caps afterwards. Try the following lines after typing in the above PROC:

scan

DIM text\$ OF 20 //Power Driver only needs this line

text\$ = "This is a TEST"

lowercase(text\$)

print text\$

This is a TEST <-- comal responds with this

It didn't change! That's because lowercase used an alias parameter. Now go back and type REF before string\$ in the PROC header so it looks like this:

```
PROC lowercase(REF string$)
```

Re-type the 4 lines above, starting with scan (or just cursor back up to them and press RETURN if they are still on the screen). Now the result is:

this is a test

This time it worked. By using the REF keyword you are telling COMAL to reference the actual variable passed to the PROC instead of using an alias. In this case, text\$ BECOMES string\$ as it is passed to the PROC, so every reference to string\$ inside the PROC actually applies to text\$.

By using the REF keyword you are telling COMAL to reference the actual variable passed to the PROC instead of using an alias.

Another difference between alias and reference parameters is that you can use literal strings in the calling statements of alias parameters, but not in REF ones. In the first example, typing:

```
print count'spaces("Hello there")
```

will return 1. However, you cannot type:

```
lowercase("Hello There")
```

after you added REF to the last example; the parameter in the calling statement must be a variable, because COMAL needs to change it if any changes are made to it inside the PROC. If it is a literal string there is no variable to change.

Hopefully this helps explain the usefulness and versatility of parameters with PROCs and FUNCs.

C64 Printer Tips

Our COMAL Support on People Link is valuable. Bring up a topic and watch what happens. Just sit back and read the questions and answers from others! Jump in and offer some help to someone who posts a question. Thus, the more of you who take part in our People Link COMAL CLUB, the more help it is to everyone! Take advantage of the \$25 Free People Link On-Line time and join in! For instance, I posted a simple question. Look what happened in just 10 days:

Date: 8/21/90 21:20 Num: 1,294
To: ALL By : CAPTAIN C
Title: PRINTER CONTROL

I just got a question in a letter from Chet Farley. He has COMAL on his C64 (probably 0.14 or Power Driver .. he did not specify) and wants to control his printers. He has a VIC 1040S, Gemini 10 and Amdex Color 1.

I believe to do this he should use a secondary address (for each mode), plus send escape codes. To use secondary address, I think he just opens the printer himself then does the SELECT "LP:"

**OPEN FILE 255,"",UNIT 4,7
SELECT "LP:"
PRINT "stuff"
LIST
etc.
SELECT "DS:"**

I know one of the issues of *COMAL Today* has the answer to this. Can anyone out there confirm this?

Date: 8/22/90 17:40 Num: 1,299
To: CAPTAIN C (r) By : DAVID*W
Title: R#1294 PRINTERS

Is he just trying to do basic stuff like printing program listings? If so I think just specifying the correct secondary address should do it (I never was any big fan of secondary addresses!).

If he wants to do things like printing in boldface, italics, etc. then he will need to know the escape codes for each function. I'm sure there are routines in past *COMAL Today* issues for Gemini printers; I'm not sure about the others.

Date: 8/22/90 20:44 Num: 1,300
To: DAVID*W (r) By : CAPTAIN C
Title: R#1299 SECONDARY ADDRESS

David ... to open the printer using a secondary address ... was my method right? I haven't done it for several years now!

Date: 8/22/90 21:01 Num: 1,306
To: CAPTAIN C (r) By : DAVID*W
Title: R#1300 PRINTER COMMANDS

Yes, I rarely use a printer with Power Driver, but it looks right to me. Meanwhile, I just checked the COMAL Handbook. The proper syntax is:

OPEN FILE x,"",UNIT 4,7

to open printer device 4 with a secondary address of 7. The word UNIT may be optional. There are other secondary addresses to put a printer in different mode (like upper/lower case), but they probably vary by printer and you'd have to refer to the manual. But at least the above command should get the printer to print SOMETHING, even if the case needs to be reversed. I'll try it out on my printer to be sure and let you know (it's an Epson compatible).

After opening the printer, you should be able SELECT OUTPUT "lp:" and then PRINT or LIST or whatever. I think there is a problem listing the disk directory to the printer (with the CAT command) in Power Driver, but I know there is some way it can be done.

Date: 8/23/90 18:33 Num: 1,310
To: DAVID*W (r) By : CAPTAIN C
Title: R#1306 MANUALS

Yes ... But I think you HAVE to use a specific file number if you want the open file to the printer to be picked up by the SELECT OUTPUT "LP:" command. I think it was either file number 1, 254, or 255. And yes, the secondary address of 7 gives a lower case mode on some printers. Other secondary addresses have other affects ... the user would have to check the manual that goes with their printer interface and/or printer.

Continued...

C64 Printer Tips

...Continued

Date: 8/25/90 10:58 Num: 1,331
To: CAPTAIN C (r) By : DAVID*W
Title: R#1310 POWER DRIVER PRINTING

After some experimenting, I finally figured out how to get Power Driver to make the printer do what you were asking about, Len.

I was getting nowhere, even in C64 COMAL 2.0, at first. Using SELECT "lp:/s7", which is supposed to use a secondary address of 7, didn't even work. Finally I tried flipping the DIP switch on the interface to NORMAL mode, as opposed to TRANSPARENT, and the above method worked correctly. So I switched to Power Driver. Here's what finally worked.

OPEN FILE 255,"",UNIT 4,7,READ

```
// COMAL added the "READ", I didn't.  
// The 7 selects upper/lower on my interface.  
SELECT OUTPUT "lp:  
PRINT "Test"  
//prints "Test" to printer in upper/lower case  
LIST  
//lists program to printer in upper/lower case  
SELECT OUTPUT "ds:  
CLOSE 255
```

Another example:

```
OPEN FILE 255,"",UNIT 4,0,READ  
//secondary address of 0 means UPPER case  
SELECT OUTPUT "lp:  
PRINT "Select"  
//prints [reverse heart]ELECT  
LIST  
// lists program to printer in upper  
// case/graphics or whatever your printer  
// prints in place of graphics  
SELECT OUTPUT "ds:  
CLOSE 255
```

Opening file 255 with a secondary address of 3 and selecting the printer as the output device produced a hex dump of everything printed, as it's supposed to, at least with my interface.

So, once you OPEN file 255 to the printer, whatever secondary address you select will be used by any

subsequent SELECT OUTPUT "lp:" until you CLOSE file 255 and open it with a different secondary address.

All the above was done with the interface switched to NORMAL mode, but then when I switched it back to TRANSPARENT mode, it still worked. I don't know why it only didn't work right last night. Depending on your printer, you may have to put the interface in either NORMAL or TRANSPARENT mode.

I knew there was a reason I hated printer interfaces! :)

Date: 8/25/90 14:01 Num: 1,332
To: DAVID*W (r) By : CAPTAIN C
Title: R#1331 THATS IT

Thanks, David. I think that is what I was looking for. Now I can send an answer to the letter I got. Hope it can help others now as well.

Date: 8/26/90 21:34 Num: 1,346
To: CAPTAIN C (r) By : COMALTODAY
Title: R#1294 FILE 255

That's true. File 255 under Power Driver or 0.1x is the "LP:" file. If no file 255 has been opened, COMAL opens Unit 4,0 (it SHOULD have been designed to open UNIT 4,7, but too late now!).

This same trick can be used to redirect COMAL command output to a disk file. The code below sends a copy of the DIRectory to the file "TEST":

```
OPEN [FILE] 255, "TEST", WRITE  
SELECT [OUTPUT] "LP:  
DIR  
SELECT [OUTPUT] "DS:  
CLOSE [FILE] [255]
```

Date: 8/26/90 23:17 Num: 1,349
To: COMALTODAY (r) By : CAPTAIN C
Title: R#1346 PRINTER // FILE

Wait a minute, Joel ... that does not sound right ... If the file number 255 is reserved for the printer ... how did you get it to go to the disk drive (they are two different unit numbers?)

Continued...

C64 Printer Tips

...Continued

Date: 8/27/90 17:18 Num: 1,353
To: CAPTAIN C (r) By : DAVID*W
Title: R#1349 COMAL=FLEXIBILITY

I think file 255 is reserved for "lp:", not specifically the printer.

Normally when you say "lp:" you mean the printer and by default SELECTing "lp:" for the output opens file 255 (with a secondary address of 0).

But as we found out you can first open file 255 with a different secondary address, then every time you SELECT OUTPUT "lp:" you will be using file 255, with whatever secondary address it was OPENed with.

Apparently you can also open file 255 to a different DEVICE, in which case a SELECT "lp:" will still use file 255 and send the output to whatever device was specified in the OPEN line. Right, Joel?

At least that's what it looks like to me. Nice and versatile! :)

Date: 8/28/90 19:45 Num: 1,362
To: DAVID*W (r) By : CAPTAIN C
Title: R#1353 FILE

Wait a minute! Does this mean we can do a SELECT to a file with POWER DRIVER? Did anyone ever try it?

Date: 8/28/90 22:09 Num: 1,367
To: CAPTAIN C (r) By : DAVID*W
Title: R#1362 SELECT OUTPUT TO DISK
WITH POWER DRIVER -- YES!

You learn something new all the time!

I played around a little more with Power Driver and the SELECT OUTPUT command. You *CAN* select output to the disk drive, using the "trick" that Joel mentioned.

It seems file 255 is reserved for the name "LP:". Usually that means the printer, but you can override that. Here's what I did...

OPEN FILE 255,"test",UNIT 8,1

[You can also say WRITE instead of UNIT 8,1]

This created the file "test" on drive 8. I think the ",1" opens the file in WRITE mode. I left it off the first time and the drive light started blinking. I thought I remembered from way back that ",1" opened the Datasette in WRITE mode, so I tried it here and it seems to work. After opening the file, I tried some printing...

PRINT "hello"

// This printed "hello" to the screen.

SELECT "lp:"

// LP gets its info from file 255.

PRINT "hello"

//this time nothing was printed on screen.

LIST

// Nothing printed to screen,
// but the drive started spinning.

SELECT "ds:"

// Back to the screen.

CLOSE 255

Now when I examined the file "test", I found that it contained the word "hello", followed by a listing of the program that was in memory! It actually WORKED! (Why was I surprised? :))

Next I tried opening the file with a ",0" instead of ",1" on the end, presumably READ mode. However, when I used SELECT "lp:" that time nothing happened. Apparently SELECT OUTPUT is ONLY for output and can't be tricked into inputting.

I repeated the above procedure with a file number other than 255 and it didn't work. Obviously, file 255 and "lp:" work together.

Now there is an easy way to send a Power Driver program's output to either the screen, the printer, or the disk drive, depending on what the user wants to do, and the only change that needs to be made in the program is in the OPEN line! (Actually this method always worked -- I just wasn't aware of it.)

Continued...

C64 Printer Tips

...Continued

Date: 8/28/90 23:21 Num: 1,368
To: DAVID*W (r) By : CAPTAIN C
Title: R#1367 SHARE

Thanks for the report DW! And thanks to JOEL!

Hey ... what other secrets and tips do users out there have to share with us?

Date: 8/26/90 21:38 Num: 1,347
To: DAVID*W (r) By : COMALTODAY
Title: R#1306 FILENAME STRING

The filename in the OPEN statement for a printer gets sent to the printer as if it were in a PRINT statement (but it must be a single string expression). Here is an example:

OPEN FILE 255,CHR\$(27)+"E",UNIT 4,5

For an interface which defines Secondary Address 5 as a "Transparent Mode w/ ASCII Conversion" which is hooked up to an Epson-a-like, the above statement will not only OPEN the printer with Secondary Address 4, but will also send the command to activate Enhanced (pseudo-boldface) mode!

Date: 8/26/90 23:15 Num: 1,348
To: COMALTODAY (r) By : CAPTAIN C
Title: R#1347 FILENAME

Yes, Joel ... thanks for jogging the memory (mine that is). I recall that the command as a file name is sometimes quite useful, too.

Date: 8/28/90 19:50 Num: 1,363
To: COMALTODAY (r) By : CAPTAIN C
Title: R#1346 4,7

You mentioned: "COMAL opens Unit 4,0 (it should have been designed to open 4,7, but ...)"

I am pretty sure that 4,7 *IS* the default for Power Driver. COMAL 0.14 had the 4,0.

Date: 8/28/90 20:07 Num: 1,364
To: CAPTAIN C (r) By : DAVID*W
Title: R#1363 0 OR 7?

I don't know. When I was experimenting I only got upper case until I opened the file with a sec. address of 7. At least it's easy to change.

Date: 8/27/90 17:26 Num: 1,354
To: COMALTODAY (r) By : DAVID*W
Title: R#1347 OPENING PRINTER

Thanks, Joel. That adds even more flexibility than I thought!

Date: 8/30/90 22:44 Num: 1,396
To: CAPTAIN C (r) By : COMALTODAY
Title: R#1349 FILE#, NOT UNIT#!

FILE number, not UNIT number!

PowerDriver and 0.14 use FILE number 255 as the "LP:" channel. That's why you're not supposed to use that number for your own files in ordinary circumstances. (0 and 1 are also reserved for LOAD and SAVE, respectively.)

If you do a SELECT [OUTPUT] "LP:", COMAL checks to see if a FILE 255 is already open. If so, it uses it. If not, it does the equivalent of an:

OPEN [FILE] 255, "", UNIT 4,0, WRITE

and uses that until SELECT [OUTPUT] "DS:" which closes it. By OPENing FILE 255 beforehand, you can send "LP:" output anywhere you want!

Date: 8/30/90 22:58 Num: 1,399
To: COMALTODAY (r) By : CAPTAIN C
Title: R#1396 7

Yes ... that sounds right ... for COMAL 0.14. However, Power Driver I think changed the default sec address from 0 to 7. Thus UNIT 4,7 would be used. True?

Continued...

C64 Printer Tips

...Continued

Date: 8/31/90 17:43 Num: 1,404
To: CAPTAIN C (r) By : DAVID*W
Title: R#1399 DEFAULT 0

No, Power Driver still seems to use a default secondary address of 0 with SELECT OUTPUT "LP:". When I used that line the printer printed in upper case. It wasn't until I opened file 255 with a secondary address of 7 that I could get upper/lower text on the printer.

At least, now that I know how to do it, it is no problem.

Date: 8/31/90 22:47 Num: 1,414
To: DAVID*W (r) By : CAPTAIN C
Title: R#1404 LOWER

DW ... are you sure?

I am nearly certain that David Stidolph told me he changed Power Driver so it defaulted to lower case on the printer.

Does it depend on your printer interface?

Date: 9/01/90 13:38 Num: 1,415
To: CAPTAIN C (r) By : DAVID*W
Title: R#1414 INTERFACE

That could be. The interface may make some changes that affects upper or lower case.

Date: 8/30/90 23:13 Num: 1,400
To: DAVID*W (r) By : COMALTODAY
Title: R#1367 MORE ON FILE 255 WITH PD

Listen up, guys! This trick works for ANY ",WRITE"-mode OPEN on FILE 255! It should also work with ",APPEND!" You do NOT need to specify ",UNIT 8,1" on disk files to Unit 8, since you don't NORMALLY have to specify them!

The following all work fine, and re-direct SELECT "LP:" accordingly:

OPEN 255,"myfile",WRITE
OPEN 255,"",UNIT 4,5 //Transparent mode?
OPEN 255,"myfile,p",WRITE //PRG file!

OPEN 255,"myfile,u",UNIT 9,WRITE

//USR file on Unit #9!

OPEN 255,CHR\$(40)+CHR\$(97),UNIT 2,WRITE

// Serial port, 1200bps, 7 bits, Even parity,
// 1 stop bit, hardware handshake!

With that last one, you could (with a Hayes-compatible modem) dial a friend with something like:

PRINT FILE 255: "ATDT555-5555"

and then LIST the program to him! (After a SELECT "LP:", of course...)

Date: 8/31/90 0:08 Num: 1,403
To: COMALTODAY (r) By : CAPTAIN C
Title: R#1400 GREAT STUFF

This is getting interesting ... We want more, JOEL!

I don't recall seeing some of these "tricks" before! And C64 COMAL has been out for years! Thanks, Joel.

Date: 8/31/90 17:46 Num: 1,405
To: COMALTODAY (r) By : DAVID*W
Title: R#1400 MODEM

I found out that opening file 255 in WRITE mode, as opposed to using UNIT 8,1, works also. That last example you gave is an amazingly easy way to send data over the modem! Thanks for the tips.

FISH DISK CORRECTION:

**Tetris, Tetrix, Steinschlag,
Quattro and Obsess games
have been removed from disks
173, 221, 230, 238, 305 and
324 at the request of Spectrum
Holobyte and Fred Fish. Thus
they are crossed out on the
back cover Reference List.**

Amiga Fish Disks - A Goldmine of Programs

Ten more disks were released after the back cover reference list was created. Here they are:

Fish To Go #391

Curses v1.10 Eco v3.40
FractalLab v1.0 ListPlot

Fish To Go #392

BTNTape (Better Than Nothing) v1.0
CPlot v1.0 Pmode
SetNoClick Spades v1.1

Fish To Go #393

FileIO v1.9 FontConvert v1.0
FuncLib ILBMLib v0.3
LibTool PrintSpool v0.1
RexxIntuition RexxLib

Fish To Go #394

Aniptrs 'Liner v2.00
Pics PrintImage v1.0

Fish To Go #395

(removed due to copyright problems)

Fish To Go #396

ColorCatch v1.0 NewLook v1.0
PBar v1.0 PCalender v1.0
PClock v1.0 PFiler v1.0
Resident v1.0 RoadRoute v1.6
TurboTopaz v1.0

Fish To Go #397

DKBTrace v2.0

Fish To Go #398

DClock v1.29 Formatter v2.7
GMC v9.6 HunkFunk
KeyMacro v1.6

Fish To Go #399

AutoCLI v1.6 CCLib v3.0
PrettyWindowsTrackDisplay

Fish To Go #400

DriveWars v1.0 ParNet v2.4
ReqLib SetCPU v1.6
SF2 v2.0

There are now 400 disks in the freely re-distributable library of Amiga programs and aids, fondly known as **FISH DISKS**. If you have an Amiga, you should check into this collection. You will find wonderful utilities, great games and fantastic application programs. And it costs you only \$1.95 per disk (*and the disks are generally packed full*). Jesse Knight used AmigaCOMAL to create a quick list of programs on the first 390 disks (*see the back cover.*) This is the best reference list of Fish Disks available anywhere! However, it is not much help if you don't know what the programs are. What you need is a summary of each program. That would take up too much room to print here, but this list is available on disk (as text files) for just \$1.95 (*it's called the Fish Catalog Disk*).

You can order **Fish Disks** from COMAL Users Group ... and our price of \$1.95 each is the lowest we've seen (other places charge between \$3 and \$6 each). Plus you are helping to support COMAL by getting them from us. Here are just a few of the programs:

ASDG-RRD (disk#241) - A recoverable ram disk, fondly known as VD0: ... you can warm reboot your Amiga, and anything in VD0: will still be there! You also can configure it to be the same size as a floppy disk, and then use DISKCOPY!

PowerPacker (disk#253) - Compresses executable programs in a way that they automatically expand when executed. Saves precious space on WorkBench disk!

Snap (disk#326) - Grabs text or graphics from the screen into the clipboard. Use it with your modem terminal program to "re-send" text from your review buffer.

SID (disk#338) - Fantastic Disk Utility. Lets you point and click to copy files, read text, edit, change attributes and more. It even is smart enough to unpack LHARC, ZOO and ARC files, automatically for you! A dream program that we couldn't be without. Has its own configuration file you can change. Highly recommended!

Drip (disk#347) - A great game for kids (ages 3-99) with modern music and voices (try it with stereo speakers). I haven't passed level 3 yet, but it's fun trying.

LHarc (disk#383) - This program is required if you download programs from PLINK or other networks. It compresses / un-compresses files. When compressed they take up less disk space, and take less time to transfer from a network.

DMouse (disk#258) - Screen blanker, mouse pointer blunker, popup CLI, auto window activator, mouse accelerator and much, much more!

PPrefs (disk#242) - Replaces the preferences program. It takes up less space on your disk plus lets you set up your screen to have more than 25 rows and more than 80 characters per row. We have ours set for 86 characters on 28 lines!

Aquarium (disk#301) - A database of Fish Disk program info. **Disk #369** updates the data (you need both). Great in conjunction with our Fish Catalog Disk.

SPECIAL OFFER: Get the Aquarium set (disks #301 & #369) and we will toss in a ***FREE*** Fish Catalog Disk! These disks will help you find many programs that you need! You may wish to get other Fish Disks from the list above at the same time to give you a feel for this great library of disks! [see also pages 4 & 9]

AmigaCOMAL 2.04: The Changes

In version 2.04 some important changes and improvements have been made, including:

- more installation options (including overscan)
- new keyboard layout
- improved editing facilities
- a new file requester
- the automatic creation of a SAVE file icon is now an installation option
- names in a COMAL package can be exported using another name (an alias)
- new improved graphics and turtle packages
- better access to IO-system

Changes and improvements are described below, except the new graphics and turtle packages, which are described in a special file named *Graphics.doc* on the AmigaCOMAL 2.04 update disk.

1. INSTALLATION

Some new installation options are added to the installation program:

1. You may choose if the command window should be brought to the front when a program ends or if the execute window should stay in front.
2. You may choose if AmigaCOMAL should start in insert mode or in overwrite mode.
3. The typing mode (insert or overwrite) may be toggled by pressing the «Ins» key. But you may choose if you want to stay in the current typing mode or return to the startup mode (chosen in 2. above) after «Enter» or «Esc».
4. You may choose if an icon should be created automatically each time a program is saved with the SAVE command. The choice can at any time be suppressed by the new SAVE parameters (*see next page*).

The installation program is provided with a context sensitive help function accessed by pressing the «Help» key.

The values chosen in the installation program are stored in a file named AmigaCOMAL.preferences (unless you chose some other name). At startup, AmigaCOMAL searches for this file in the following directories (in this order):

1. The current directory
2. The directory containing AmigaCOMAL itself
3. DEVS:

2. NEW KEYBOARD LAYOUT

The editor keys have been changed to be more compatible with other Amiga software and new editing facilities are added.

Summary:

«Shift» + «CursorRight»	Go to end of line.
«Shift» + «CursorLeft»	Go to start of line.
«Shift» + «CursorUp»	Go to top of window
-or- «PgUp»	or if already on top and the top line is a program line, the previous page is listed.
«Shift» + «CursorDown»	Go to bottom of window
-or- «PgDn»	or if already on bottom and the bottom line is a program line the next page is listed.
«Alt» + «CursorUp»	List first page of program.
-or- «Home»	
«Alt» + «CursorDown»	List last page of program.
-or- «End»	
«Ins»	Toggle insert mode.
«Alt» + «Del»	Delete line.
«Alt» + «Ins»	Insert line

Note that «Ins», «PgUp», «PgDn», «Home» and «End» are found on the numeric keypad and that the «Shift» key must be used in addition to those keys.

3. THE EDIT COMMAND

An EDIT command has been introduced. This command lists the first page of the program and puts you into edit mode.

While in edit mode you cannot leave a line that is not a correct AmigaCOMAL line. Each time you try

Continued...

AmigaCOMAL 2.04: The Changes

...Continued

to leave a line by using one of the keys described in the previous section, the line is scanned, and if correct it is put into the program.

Thus, while in edit mode, the lines you see in the command window are the same as those in the program area. The only way to leave a line without putting it into the program is to use the mouse, press «Esc», or execute a command from the drop down menu. The RENUM command is an exception to this, allowing you to make more room between two lines while remaining in EDIT mode. The lines are renumbered in the program and on the screen.

In addition to this, «Alt»+«Del» deletes the cursor line both in the command window and in the program itself. «Alt»+«Ins» inserts a line with a line number between the surrounding lines and at the same time put this line into the program itself.

4. NEW SAVE PARAMETERS.

It is possible to choose if an icon should be created for a SAVE file.

The command:

SAVE "name",i

will always create an icon and the command:

SAVE "name",n

will not create an icon. Both commands are independent of how AmigaCOMAL was installed.
The command:

SAVE "name"

will create or not create an icon, depending on how AmigaCOMAL has been installed.

5. NEW FILE REQUESTER

A file requester with many of new facilities has been added.

This file requester works with the LOAD, SAVE, ENTER and LIST commands. To activate it with the ENTER command type:

ENTER ""

or hit «F8» followed by «Enter»

Likewise for LIST.

6. EXPORT ALIAS IN COMAL PACKAGES

It is now possible to EXPORT a variable, a procedure or a function in a comal package with another name by using the AS keyword. Example:

EXPORT alfa() AS beta
EXPORT fd(), fd() AS forward

The first example exports procedure **alfa** using the name **beta**. The second example exports the procedure **fd** as both **fd** and **forward**.

The new Turtle package has many examples of this new EXPORT facility.

7. NEW COMAL PACKAGE EXTENSION

The extension of a comal package (*a package that is written in COMAL*) is changed to .cmp to make it possible to tell if a package is a comal package or a code package (*compiled in ASM or C*).

The USE command/statement:

USE name

will now search for the file until one is found in this order:

name.cmp
Packages/name.cmp
name.pck
Packages/name.pck

8. NEW PACKAGES

In addition to the packages discussed in the manual the following packages are added:

POTGO_LIBRARY

Machine coded package to be used as an interface to the Amiga Potgo routines.

IFF_LIBRARY

Machine coded package to be used as an interface to the public domain library **IFF.library** which is supplied with AmigaCOMAL.

To get a quick list of commands in a package:

USE packname
LISTPACK packname

To have this list go to the printer (or a file):

LISTPACK packname TO "location"

9. Extended COMAL Structure & IO-Structure

A new field has been included in the Comal Structure. It now looks like this:

```
RECORD comalstructure@
FIELD sp_top#, sp_bottom#
FIELD pck_link#
FIELD taskid#
FIELD dosbase#, intbase#, gfxbase#, laybase#, fontbase#, iconbase#
FIELD eventflag!, eventcount!
FIELD comwinsig!, comkbdsig!
FIELD excwinsig!, exckbdsig!
FIELD serialsig!, timersig!
FIELD verno! // Current AmigaCOMAL version number
FIELD res1!, res2%
// Call comalwait with comwinsig! or excwinsig! and the next
// field will be pointing to a list of cmlintuimessages (the
// real content of IntuiMessages).
POINTER cmlintuimessage@ TO cmlintuimessage@
POINTER active_io@ TO io_struct@
POINTER command_io@ TO io_struct@
POINTER exec_io@ TO io_struct@
ENDRECORD comalstructure@
```

The new field is the pointer to **cmlintuimessage@** which is part of the **IntuiMsg** sent by intuition. This structure looks like this:

```
RECORD cmlintuimessage@
POINTER nextmsg@ TO cmlintuimessage@
FIELD class#
FIELD code%
FIELD qualifier%
FIELD iaddress#
FIELD mousex%, mousey%
FIELD seconds#, micros#
FIELD idcmpwindow#
ENDRECORD cmlintuimessage@
```

If **comalwait** is called with either **comwinsig!** or **excwinsig!** signal bit set, this pointer in the comalstructure will be set to point to a list of cmlintuimessages - one message for each **IntuiMessage** sent by Intuition.

The **IO_struct** now includes pointers to the **IO-Message** and **IO-Reply** port used by AmigaCOMAL in communication with the **Console.device**. A copy of the **IO-Message** can be used to change the keyboard layout.

The **IO_Structure** now looks:

```
RECORD io_struct@
FIELD screen#
FIELD screentype%, screendepth%, screenwidth%, screenheight%
FIELD window#
FIELD windowdepth%, charno%, lineno%
FIELD gzxoff%, gzyoff%
FIELD windowwidth%, windowheight%
FIELD fontid#
FIELD fontheight%, fontwidth%, fontbase%
FIELD virtual#
FIELD cursor!, softstyle!
FIELD menuhd#, menybytes#
FIELD privat1#, privat2#, privat3# // Don't touch!
FIELD conmsg# // Console input message // Not active_io
FIELD conreply# // Console reply port // Not active_io
ENDRECORD io_struct@
```

Note that the extended structure cannot be used with the **Active_IO** - only **Command_IO** and **Exec_IO**.

10. MISCELLANEOUS

In the **REPORT** statement, a text may be added, for instance:

REPORT 899, "A very bad error!"

If the specified error number has no error text the specified text will be the error text.

In the text, constant codes (equivalent to **CHR\$**) may be included by writing the code preceded by a double quote ("), for instance:

PRINT "Oh no! "7" // Flash the screen

[There are some bugs in this update. A summary of them will be posted on PLink and sent to the author in Denmark. We expect another update as a result.]

AmigaCOMAL 2.04 Update Notes

by Svend Daugaard Pedersen

[These notes just arrived from Svend, the author of AmigaCOMAL].

Enclosed with this letter is the very latest version of AmigaCOMAL - version 2.04. On the disk you will find both the interpreter and the development part (including the compiler). That is how we are going to do it in Denmark to speed up the sales. You get the compiler and all the development stuff for no extra cost. Maybe you could do the same.

(*Yes, that is fine with us. That will be our special offer: buy AmigaCOMAL - get the compiler and development kit free. We also will make the update available to registered users for just \$10.*)

Version 2.04 runs under WorkBench 2.0 both on the Amiga 3000 and the Amiga 500/2000.

Unfortunately, I have not had the time to make all the nice packages requested, but some important changes and additions are made, especially with the graphics packages. UniGraphics (formerly called PCgraphics) and the Turtle packages have been changed and extended such that they are now 99% compatible with the UniComal IBM PC COMAL packages.

Here are some notes in regards to error reports and suggestions made by Jesse Knight:

(1) `^ptr():=^a!()`

I have to admit that the error message is not very informative (I think the errors are some of the weak points in my AmigaCOMAL). In this case the message comes because the line should read:

`^ptr:=^a!()`

In the statement `^ptr!` acts like an integer variable and that is why the parenthesis should not be there.

(2) `src$():=spc$(80)`

The error has been fixed (and I hope it has not introduced new errors).

(3) `scroll` bug is fixed.

(4) `Except` bug

`Except` has not been tested very much and unfortunately I have lost the source. That's why I cannot help much. But I saw that you are sending signals to yourself. I don't think it is possible to send a signal to your own task. Is this the reason for the error?

(5) Some (but not all) of the suggestions have been included.

I hope you like this new version with all its small changes.

2.04 Update Via People Link

We are trying out a new idea ... offering the \$10 update to 2.04 for free if you download it from People Link (pay only the connect time). This of course applies only to registered AmigaCOMAL users. We are setting up a special section inside the COMAL Club on People Link for update distribution. You must send EMail to Captain C and to David*W requesting access to the UPDATE section, since it is a restricted access section.

It is easy to send one EMail to two or more people on People Link. While in the COMAL Club, type: /MAIL. Send it to: Captain C. While typing in the message (or after you type /END) just type: /CC David*W.

That will send a carbon copy of the EMail.

You must have sent in your registration sheet for AmigaCOMAL in order to be allowed access to the special UPDATE section.

In exchange for the FREE update, we only ask that you post your comments on it in the NOTICE area of the COMAL CLUB.

Bug Reports: If you find a bug in AmigaCOMAL, please let us know right away by posting a notice on PLINK or via US Mail. We will forward these reports to Svend in Denmark. He appreciates our feedback on AmigaCOMAL.

UniComal

The 3.0 Language DOS, OS/2 and UNIX

UniComal users have the option of moving programs written in UniComal between computers with different operating systems, including UNIX. When UniComal is used under OS/2 and UNIX, several programs can run at the same time, since OS/2 and UNIX are multi-tasking operating systems.

UniComal is a programming language system which has been hand-crafted for specialists. It is designed (and used extensively) by experts in their respective fields who do not have time to become master system programmers. These professionals can satisfy their requirements for accessible computing power and useful results with UniComal. Furthermore, they can achieve this quickly in a user-friendly programming environment.

UniComal combines the user-friendliness of an interactive programming language with the speed of a compiler. Nearly identical programming environments are provided for DOS, OS/2 and UNIX. A powerful interpreter permits UniComal programs to be developed interactively - quickly and efficiently. Instant checking of individual program functions, procedures and modules and of overall program structure greatly assists development of error-free programs.

A tested program can be compiled to create an executable file, which can then be run by any user without the aid of the UniComal system.

UniComal version 3.0 is an Object Oriented Programming language (OOPs) which allows the use of structured variables, where data and the operations performed upon them can be associated together in one unit.

Extended Record and Type Concepts

A new and revolutionary concept has been introduced in UniComal version 3.0: STRUC. This is a genuine data structure where it is possible to define operations on user-defined, structured variables (records). This makes simple, user-friendly object oriented programming accessible.

Modules in UniComal, Machine Code and C

The module concept has been implemented (a module is an improvement over the "package" of COMAL 2.0). One can write one's own modules and structures ... or incorporate modules developed by others into a program. Modules can be written in Comal, C and machine code (C-interface to DOS and OS/2 available January 1991). Modules written in Comal can be either external or internal. This makes it possible to prepare highly modular programs, making UniComal programs easier to maintain and update. The new module concept also provides the option of defining static, local variables. External modules and structures can be dynamically linked to the main program. Thus revisions can be made independently in the main program and in modules without the need to subsequently compile the entire program system again.

Pointers

Pointers and pointer variables have been introduced in version 3.0. It is possible to define dynamic data structures with the inherent advantages of more efficient memory use.

64K Barrier Broken

The 64K limit for data has been eliminated. In addition, programs can consist of an arbitrary number of modules, each of which can be up to 64K in size. In OS/2, there also is no 640K limit - the entire virtual storage area can be used for programs and data.

Event Handling

UniComal 3.0 can handle external events. For example, a program could execute certain routines (procedures) every time an external device issued an interrupt. It is possible to define the priority of events. In connection with control tasks, data collection, etc., this facility is very important to the effectiveness and maintainability of a program.

Features of UniComal 3.02 are shown in the chart on page 26.

IBM PC COMAL 3.02

Entry Level

This version is specially designed for users who do not need the extra facilities which are built into the *Standard* level and higher. It is an interactive programming language which is easy to work with.

One can use up to 64K for arrays and simple variables (*Standard* level and higher storage is limited by available memory).

One can use up to 8K for heap (dynamic storage) (*Standard* level and higher heap can be expanded until there is no more available memory).

One can work with internal modules (packages) and structures written in COMAL. One also can use external modules (packages) written in another language (*Standard* level and higher allow internal and external modules (packages) and structures written in COMAL or in another language).

External interrupt handling is not supported with the *Entry* level.

80x87 math co-processor is not supported with the *Entry* level.

A compiler is included that will transform *Entry* level COMAL programs into an executable (.EXE) file. However, this compiler will not work on COMAL programs written with the *Standard* level or higher.

OS/2 support is not available for *Entry* level.

IBM PC COMAL 3.02 Prices (includes shipping)

\$235.00	Entry Level
\$435.00	Standard Level
\$635.00	Extended Level
\$835.00	Developer Level
\$300.00	SCOM Module
\$ 50.00	UniNet Module
\$125.00	UniWindows & UniMenus Modules

IBM PC COMAL 3.02

Standard Level

Only the users imagination will place limits on what can be developed with this complete interactive IBM PC COMAL. This user-friendly programming language is modular and object oriented, which makes it possible to make optimum use of available hardware.

One can (if you wish) limit the amount of storage to be used for arrays and simple variables. One can write modules (packages) and structures (RECORDs) in COMAL and work with modules written in Assembler and C in a fast, interpreter-based environment. Two of the many advantages with interpreters are development speed and dependability. UniComal saves you development time and enhances security.

Programs written in UniComal are easy to move. If you develop a UniComal program under OS/2, it can be moved with only minor changes to DOS or UNIX. It can then be compiled for use on these operating systems.

One can make a service agreement (direct with UniComal) on all UniComal products (except *Entry* level). This provides the right to direct telephone service from UniComal. This service agreement also provides the right to receive free SPE (Small Program Enhancement) updates. This means regular updates with improvements and enhancements.

UniMouse module is included. This makes it possible to work with a mouse. The programmer can read the position of the mouse and the mouse buttons from within a COMAL program.

Btrieve Demo is included. This demo illustrates the use of UniComal's interface to Novell's Btrieve file management system, where it is possible to construct fast, flexible databases with many different search keys.

The compiler is not included (*if you need to create stand-alone executable files, the full compiler is included beginning with the Extended version*).

IBM PC COMAL 3.02 Extended Level

This includes the complete interactive IBM PC COMAL and includes the full *Standard* level with many extensions. The UniComal service agreement is available (as mentioned previously).

The full UniComal 3.0 compiler is included, which transforms a UniComal program into an DOS executable (.EXE) file for distribution.

As with the *Standard* level, UniComal programs can be moved between OS/2, DOS and UNIX.

SCOM is included. SCOM is the Serial COMmunications module. It allows you to control several RS-232 ports at once. When you want to control external units, modems, etc., this module makes the job much easier.

UniMouse module is included. This allows you to work with a mouse (same as with the *Standard* level). The Btrieve Demo program is also included.

UniComal For UNIX Notes

UniComal for UNIX is supplied under different conditions than for DOS and OS/2. It currently runs on the SUN 3/50, 3/60, 3/80 under BSD 4.2.

Screen handling differs since the extensive DOS and OS/2 character set is not available. The current version of UNIX only supports the 128 ASCII character set, thus excluding, for example, special national characters and line drawing characters. Consequently, a program which use boxes or foreign characters cannot run under the UNIX version without some revision.

The file systems under the different operating systems are similar. But UNIX uses a / slash instead of a \ backslash as a separator in paths (similar to AmigaCOMAL).

UniComal's standard DOS modules are accessible to the UNIX version only to a limited extent.

Information about and ordering of UNIX COMAL is done directly with UniComal A/S, Tvaermarksvej 19, DK-2860 Soborg, Denmark

Telephone: +45 31 67 35 11 FAX: +45 31 67 63 14

IBM PC COMAL 3.02 Developer's Level

This is a complete interactive programming tool which can run under three different operating systems (DOS, OS/2 and UNIX). The package recommends itself particularly to industrial customers due to its interactive environment and its ability to communicate with other languages such as C or Assembler. UniComal 3.0 is an object oriented programming language (OOP) which opens completely new possibilities for easing the creation and maintenance of program systems.

An included compiler transforms a UniComal program into an executable (.EXE) file for distribution. One can develop a comprehensive system under OS/2 and move the complete program to DOS or UNIX and compile it under the relevant operating system. No royalty need be paid to UniComal for programs prepared in this way.

SCOM and UniMouse modules are included, as with the *Extended* level.

The UniComal Btrieve interface is included. This module makes it possible to communicate with Novell's Btrieve file management system, where one can build up flexible and fast databases with many different search keys. One must already have the Btrieve system itself, or it may be purchased separately.

C Development Disk is included. It contains a C-interface to UniComal. With this interface one can write modules in C and use them from UniComal without having to worry about the language in which the module was prepared.

Assembler Development Disk is included. It contains an Assembler interface to UniComal. By means of this interface one can write modules in Assembler and use them from UniComal. This facility makes it practical to prepare machine related routines in an Assembler module and to work interactively with the system at machine level.

OS/2 support is included.

UniComal 3.02

IBM PC COMAL 3.02 is now shipping. This article briefly summarizes the changes in the upgrade from version 3.01.

Save Warning - If the program has been changed, a warning is given if you issue a command that would remove the program from memory (BYE, NEW, LOAD and ENTER).

Editing Modules - In version 3.01 you were allowed to edit an external module that was loaded with the main program. However, these changes could not be saved. This caused problems later. Version 3.02 still does not allow the saving of an external module loaded with a program (*that capability will hopefully be added in a future release*). However, 3.02 now does not allow you to edit the module directly. You must use ALT-M to load in the module for editing (*your main program is saved by this command too*). Then just run your main program again.

Sharing Modules - In version 3.01 a module could be shared (ie, the same external module file could be called by TIMER1 and TIMER2). Very few users actually used this capability, but many were confused by it. Therefor, beginning with version 3.02 an external module can only be called by a program once. If you need to have the module used twice, simply make a second copy of the file with a different name.

Line Listed - If program execution is stopped via the CTRL-BREAK key, COMAL lists the line that was executing at that time.

Profile - Profile parameters for modules can now be set in either the COMAL profile file or the DOS environment.

FIND and CONTinue - Previously, if you stopped a program and then used the FIND command, if you hit ENTER, you could not CONTinue program execution (*program has been changed, due to entering a line, even though the line itself was not changed*). Version 3.02 no longer has this problem. After a FIND command, you may use the CONTINUE command.

Reusing Modules - Modules already loaded in memory are no longer reloaded when a RUN or

SCAN command is issued. This saves time.

Use - Use is now an active statement that issues an error message if it fails. This can now be TRAPPED.

Cursor Position - When editing a program with lines that wrap onto the next line, the current cursor position is restored when cursoring up or down through the continuation line.

Blocks - Now when you refer to a block of statements (such as a procedure, function, struc or module), any preceding comments are considered part of it, including blank lines. Thus you may LIST a procedure and also see the preceding comments.

Format - A format string can now be as long as any string (previously it was limited to 255 characters).

VGA Graphics - The graphics modules now test for the presence of a VGA adaptor. Previously it only tested for an IBM PS/2 model.

Polygon Fill - 26 different fill patterns are now available for POLYGON. A style is chosen with the statement: POLYGONSTYLE(style#)

New INQ values - The clip coordinates can now be queried via the INQ function.

Text Window - The current position of the text window can now be queried with the new statement: GETTEXTWINDOW(top#,bottom#,left#,right#)

WOW! MENUS & WINDOWS!

Menus and Windows modules are now shipping. Very professional, yet extremely easy to setup and use. Price is \$125.

IBM PC COMAL 3.02 Prices (includes shipping)

\$235	Entry Level
\$435	Standard Level
\$635	Extended Level
\$835	Developers Level

\$125	Menus & Windows Modules
\$300	SCOM Module (Serial Communications)
\$ 50	UniNet

3.02 Facility	Entry	Standard	Extended	Developer	Description - UniComal 3.02
Interactive System	○	☺	☺	☺	A dialogue oriented interactive development system.
Compiler	○		☺	☺	Ability to compile to DOS .EXE file.
OS/2 Support				☺	Identical development system under OS/2.
Event Handling		☺	☺	☺	Ability to handle external events (interrupts).
Assembler Development				☺	Development tools for writing modules in assembler.
C Development				☺	Development tools for writing modules in C.
80x87 Support		☺	☺	☺	Utilization of the math coprocessor.
System Module	☺	☺	☺	☺	Module with an assortment of useful system routines.
Sound Module	☺	☺	☺	☺	Module for making sound and music.
Graphics Module ◊	☺	☺	☺	☺	Modules for CGA, MCGA, EGA, VGA, Hercules graphics.
Turtle Module ◊	☺	☺	☺	☺	Modules CGA, MCGA, EGA, VGA, Hercules turtle graphics.
UniDump Module ◊	☺	☺	☺	☺	Module for printout of graphics images on a printer.
UniTimer Module		☺	☺	☺★	Module for generation of timer events (simple multitasking).
UniMouse Module		☺	☺	☺	Module for mouse support.
SCOM Module ◊	☺	☺	☺	☺	Module for controlling serial communication (RS-232).
Btrieve Demo		☺★	☺★	☺★	A Btrieve demo program.
Btrieve Interface Module	☺	☺	☺	☺	Module for use along with the Btrieve database tool.
Windows Module	☺★○	☺★	☺★	☺★	Module for creating windows with borders & shading effects.
Menus Module △	☺★○	☺★	☺★	☺★	Module for creating a drop down menu system.

● included ○ with limits ◊ extra cost option ◊ not with OS/2 ★ with source code △ requires windows module
 Available from: COMAL Users Group, U.S.A., Ltd., 5501 Groveland Ter, Madison, WI 53716 (608) 222-4432

Getting Used To IBM COMAL 3.0

by Tom Kuiper

I've been getting used to COMAL 3.0 since it arrived a few weeks ago. Moving from 2.2 to 3.0 is a big step. Not only have lots of new features been added, but in the process, some of the character of the language has changed. Here are some of the surprises I encountered. By the way, almost everything you need to know to avoid my mistakes is somewhere in the manuals. However, I will surely never have the time to read them completely, and so must depend on COMAL's error messages to point out my mistakes. This article reviews some experiences where this approach was painful.

COMAL got bigger!

The good news is that COMAL 3.0 allows you to assemble your favorite COMAL routines into packages (now called MODULES), something which you could only do with assembly language before. The bad news is that you may have to use this feature even if you weren't planning to.

Of course, the first program I tried to convert was my biggest one; about 2400 lines of code! I LISTed it from COMAL 2.2 to disk. (Fortunately, I had the sense to move all my COMAL 2.2 to a new directory before copying the new COMAL 3.0 into my COMAL directory.) Then, I invoked COMAL 3.0 and tried to ENTER it. Well, it got about two thirds of the way through before COMAL complained that the program was too big. Starting to convert big blocks of code into MODULES, I quickly got confused. I decided to start with something smaller, learning a step at a time.

SYSTEM is an external package!

SYSTEM is now a package which must be explicitly referenced when compiling a COMAL 3.0 program. The first time I compiled a program, I ignored the "Module files" prompt, which is where packages need to be included. The program compiled just fine, but when I tried to run it, I got the error:

MODULE/STRUC file not found in line 0010

This was a bit disconcerting because it was my first program using the STRUC feature, which lets you

define data structures. Data structures can be defined either in an external file or in the program, as I had done. So why did it want to know about a structure file?

Don't ask me how I got the insight that it might, in fact, be missing the SYSTEM package. Previously, the SYSTEM package was not in a separate .PKG file. The solution to my problem was to enter

SYSTEM.EXE

at the "Module files" prompt. Just SYSTEM will not do, because COMALC then appends the default extent of .MOD, not .EXE. Now, my program ran fine, until I tried to run it on another computer.

The compiler manual is a little clearer about that. As soon as I encountered the problem, I remembered reading that if the package is to be part of the compiled program, then it is necessary to precede the file name with &. So, the correct response to the "Module files" prompt is

&SYSTEM.EXE

Incidentally, if the package you are specifying requires a path as well, then the & goes in front of the path, like this:

&\COMAL\SYSTEM.EXE

and not after it and before the file name, as UniCOMAL's instructions imply. (The instructions say that it goes before the file name.)

Run-time Libraries

This experience illustrates a feature I had previously only read about, namely, the run-time libraries. Before now, I had only worked with subroutine libraries which are permanently combined into the executable file during the linking phase. This makes programs quite large. Furthermore, every executable program which uses this library includes it, so that a library may exist many times over in many programs, gobbling up lots of disk space.

By using a library which is not accessed until the program is executed -- i.e. a run-time library -- the

Continued...

Getting Used To IBM COMAL 3.0

...Continued

size of the executable is smaller. The first version of the program, which only ran on my computer, was 64576 bytes long. The second version, which will run anywhere, is 68176 bytes long. However, the first version is only usable on machines which have SYSTEM.EXE in the drive and directory where my system has it.

What's in that MODULE, anyway?

UniCOMAL has provided a very useful new command, called INTERFACE, that you will want to get to know right away.

Used by itself, it lists the modules and structures which are in or have been linked to the current program. I generally know that, or can find it out very quickly with FIND "USE".

UniComal has provided a very useful new command called INTERFACE that you will want to get to know right away.

When INTERFACE is followed by a module name, it lists all the routines in the module. Each routine is shown with its arguments and leading comment (//) lines, which helps to remind me how to use the routine. Also, any comment (//) lines which follow the MODULE statement and before the first EXPORT are shown.

There is one odd thing about modules, so strange that it seems to me to be a bug. I was developing a new program which uses a previously developed external module. The first time I ran this program, I got the error message that the module had not been SCANNed. This surprised me, but I thought that perhaps I had made some changes in the module and forgotten to SCAN it again. I scanned the module and all was well, so I forgot about this. Recently, it happened again. I created a new program that called an existing external module that I had been happily using the day before in another program. I hadn't touched the module. Nevertheless, COMAL insisted that it had not been SCANNed.

Furthermore, it doesn't always happen. Now that I'm on the lookout for it to happen again, it hasn't. (Maybe I'm getting feeble-minded and imagining things.) [Hopefully 3.02 has corrected this]

What's in a name?

Previously you could not simultaneously have variables called, for example, NAME, NAME\$, and NAME#. You could use NAME for only one of those. That was sometimes inconvenient, because you might want to have variables of different types which are logically associated with the same name.

COMAL 3.0 has removed that restriction ... but there is a down side!

COMAL 3.0 has removed that restriction. Here is an example: I have a file with cosmic radio source data. The file name might be kept in a string called SOURCES\$. I can now open it as follows:

```
sources#=FREEFILE  
OPEN FILE sources#,sources$,read
```

But, there is a down side! It is now very easy to type something like:

```
year=VAL(string$)
```

when what I really need is

```
year#=VAL(string$)
```

The program will run along happily using the value of YEAR#=0, blissfully indifferent to the fact that the needed value is stored in YEAR.

SCOPE will show you all the variables in your program...

Continued...

Getting Used To IBM COMAL 3.0

...Continued

To help with this problem, there is a new command called SCOPE. It will show you all the variables in your program when it was last RUN or SCANned. Unfortunately, it lists them in the order in which they occur, so that YEAR and YEAR# will be in quite different locations in the list -- not an aid in picking up errors of the type described above.

The PRINT USING comma has been defined in 3.0 as a thousands separator.

New Image String Options

I often use the image string in PRINTs to make my output more easily read. For example, I might want to tabulate some coordinates this way:

PRINT USING "###, -##": longitude, latitude

so as to get a table with entries that look like this:

118, 34
120, -35

and so on. This is now a no-no! All my old image strings using this now cause a run-time error. This is because the comma has been defined in 3.0 as a thousands separator, e.g.

PRINT USING "##,###": 64000

will give the output

64,000

The fix for the original example is to put an apostrophe in front of the comma:

PRINT USING "##', -##": longitude, latitude

A real bug!

Having learned all these things, I tried to compile a real functioning program. By then I had several MODULEs defined and dutifully included the

statements

MODULE this EXTERNAL "\dir\modname.mod"

for all of them, followed by

USE this

The program compiled beautifully. No errors. (*Blessed relief!*) It ran flawlessly. (*Hurrah, hurrah!*) And hung up my computer. (*A large alcoholic drink and early to bed!*) I referred this matter to UniComal. Fortunately, they are very helpful. They could not diagnose the problem just from my description, and their suggestion for a fix did not work. I sent them a disk with everything that I thought was relevant. UniComal studied the matter and concluded that COMAL 3.01 has a bug [*hopefully corrected in the current 3.02 release that is now shipping*]. The work-around is to remove the

MODULE ... EXTERNAL ...

statements from the program, and to modify the EXTPATH line in the COMAL.PRO file to name explicitly the subdirectories in which the MODULEs reside. This, indeed, works.

COMAL 3.0 is a major improvement over 2.0.

Bottom Line

COMAL 3.0 is a major improvement over 2.0. It is a more professional language, while retaining most of the ease of use of the earlier versions. The documentation is good for the advanced user, but probably intimidating for the novice. For the latter, I suggest Len Lindsay's *COMAL Handbook*, or one of the even more elementary texts available from the COMAL Users Group. But, be warned that there are differences and, when things don't seem to work right, check the new manuals.

NOTE: release 3.02 is now shipping and fixes bugs and adds new features.

IBM PC COMAL 3.0 Notes

Letter to UniComal ↔ from Craig Van De Grift

Thank you for continuing to upgrade Comal for 80x86 machines. I have found Version 3.01 to be an excellent product with superior documentation. I am writing, however, to point out what seem to be problems. I am using a 25 MHz 80386 IBM clone with 4 MB of RAM, an ATI VGA Wonder super VGA video card with 512 kB of video memory, and a 512K HP LaserJet IIP printer.

Problem 1: 256-color pallet color setting (serious)

SETCOLOR_CMY, SETCOLOR_HLS, and SETCOLOR_RGB do not appear to work properly with my video card. Enclosed is the programming information for my video board. The failure is difficult to describe with words. Although no crash occurs, it seems that the video card registers are not as Comal expects. When I run the sample programs CMY, HLS, and RGB on the SUPP disk, the screen flashes and changes, but clearly does not do what is intended. This problem is preventing me from doing some of what I had planned, so if practical, please FAX me a patch that I can put in with DEBUG.

[This seems fixed in version 3.02]

Problem 2: OPEN and RENAME do not use DOS PATHs and APPENDs in the same way

I used a TRAPped OPEN statement to determine if a data file already existed. Since no error occurred, I proceeded to RENAME it and got a file not found error. It turns out that the file was not in my default sub-directory, but rather in a sub-directory specified by a DOS APPEND instruction. The RENAME instruction, however, could not find it. Perhaps the solution is to clearly point out this difference in the documentation and to add a system Boolean function FILE_EXISTS *filename\$* which checks for the existence of a file in a particular subdirectory (or the default). One could TRAP a RENAME which renamed the file to its same name, but a system function would be cleaner.

[This may be an MS-DOS limitation.]

Problem 3: Long listings to the printer are timed out before they are completed

If I LIST a long program to my HP LaserJet IIP, the LaserJet takes up about 600 lines in its buffer and then, while it is printing, COMAL times out. The remaining lines of my program are therefore not printed. I now list the program in 400 line sections to circumvent this problem.

[Using the MS-DOS MODE command you can change the time outs on the printer. With this change it seems to work fine.]

Problem 4: Excessive DIMensioning seems to be done Mod 2 MB

10 DIM a\$(200) of 1000 results in "out of space for arrays" while 10 DIM a\$(2000) of 1000 does not! The crossover point is between 1527 and 1528 for my configuration where I have established 121598 bytes of strings and arrays. 118 and below was OK.

[Fixed in version 3.02]

Problem 5: The following sequence of commands makes the system crash:

```
USE SYSTEM  
textmode(4)  
10 print "hi"  
save
```

After the textmode instruction has changed the screen (or brought the textscren back from a graphics screen), the save instruction gives a "51: system error", and the system then hangs up. In other cases, the save causes the hang-up without any message. A ctrl-alt-del reboot or even a hardware reboot is necessary to restore the system.

[Fixed in version 3.02]

Problem 6: UNIDUMP does not appear to print an entire 50 line vga screen

Only the first 25 lines of a textmode(5) screen are printed by UNIDUMP as invoked by the Print Screen key.

[This may be related to MS-DOS (ie, the PrintScreen key). We tested it with version 3.02 and it works fine.]

Suggestions

Could you please consider implementing the following new features:

Desired Feature 1: Boolean function FILE_EXISTS filename\$ (low priority)

See the discussion for Problem 3, above for details.

Desired Feature 2: NAMES to give alphabetical list of names (low priority)

Since variable, variable#, and variable\$ are all considered different, the programmer needs an easy way to check for incorrectly named variables where the # or \$ were accidentally left off. I currently use SCOPE for this purpose, but an alphabetized listing would be much better.

Desired Feature 3: Super VGA graphics (high priority)

I am very eager to use the super VGA modes that my video card is capable of, specifically graphicscreen(8) with 800x600x256 colors and graphicscreen(9) with 1024x768x16 colors. Perhaps the documentation for the VGA Wonder board I am sending in this FAX will be useful.

Desired Feature 4: Gray scales for printscreens to HP LaserJet printer (low priority)**Desired Feature 5: 132 line wide textmode (high priority)**

Extension of the textmodes to 132 character wide screens is desirable. e.g. textmode(6) which would be 25 lines by 132 columns, and textmode(7) which would be 50 lines by 132 columns.

Desired Feature 6: Mouse support for graphicscreens 5, 6 and 7 (medium priority)

UniMouse support beyond graphicscreen(4) is needed. I see that the current limitation appears to be that pieces of the mouse occasionally get left on the screen and that sometimes the mouse activity will interact with the retracing.

Desired Feature 7: PAGE TO "file_name\$" (low priority)

In command mode, one often sends listings, scopes, etc., to the printer and forced paging is desirable. Use of LIST TO "PRN", etc., is very useful except that one occasionally wants to send a page feed. I use a DOS batch file and enter ">page" from Comal, but it would be nice to have the PAGE instruction be able to take a "TO file_name\$" option like SCOPE, LIST, DISPLAY, etc.

Alternatively, maybe the "TO file_name\$" on SCOPE, etc., could be made into "TO file_name\$ WITH FF", where FF means add a formfeed to the end.

Desired Sample Program: Example of an application for use on wide-range of system configurations

An example of how to write an application program which will run on the widest practical range of video cards and printers is needed. This is more than just telling the user to specify text modes and graphic modes in startpar\$. The user will want to say video board name and video memory size in K, and printer name and printer memory in K. The routine decoding startpar\$ must select the text and graphic modes, and color definitions appropriate for that hardware. It would be convenient if it set some variables like screen_columns# and screen_lines#. This *boiler plate* should be easily incorporated into an application program written by someone not familiar with hardware other than his own.

Finally

I love your Introduction to UniComal book. It is excellent! I did, however, notice two minor errors:

Page 6, the final DIM shown should be DIM pointer array_name(... , not DIM struc array_name(....

Page 13, the example given for NULL is not good since the PAUSE command now performs the function of the example statement.

Thank you for your highly professional product. I look forward to further upgrades.

Compatibility

by David Warman

One of COMAL's better features is that it is highly compatible between computer systems. Recently I have had the opportunity to learn just how compatible it is.

I've been writing programs in COMAL for about six years, using COMAL 0.14, Power Driver, and C64 COMAL 2.0 for most of that time. 0.14 and its replacement, Power Driver, are introductory versions, and lack some features of the full 2.0 versions, but even so it is still remarkably easy to transfer programs between them, especially going "up" from Power Driver to 2.0. However, even though these are different versions of COMAL, they all run on the C64.

I recently bought an Amiga and AmigaCOMAL, and have had the chance to test the compatibility between some of the 2.0 level COMAL's. I can report that COMAL is indeed as compatible as it claims to be!

Since getting AmigaCOMAL I've transferred several old C64 programs to the Amiga. In fact, getting them from a 5½" C64 disk to a 3½" Amiga disk can be the hardest part. The first step is to load the program into COMAL on the 64 (or whatever computer you are using). Then store it on disk in a form that can be read by other COMAL's. Each version of COMAL SAVEs its programs in a compact tokenized binary file that is unreadable by any other implementation of COMAL. But each can also store programs in standard ASCII format with the LIST command. They can then be ENTERed into any other version of COMAL.

Commodore is unique in that it uses its own special variation of ASCII, called PetASCII. This can cause problems when transferring programs to other computers, but C64 COMAL 2.0 handles it easily; simply append the "a" attribute to the file name. For example:

LIST "filename/a+"

will list the program to disk and perform automatic translation to true ASCII. You can also perform ASCII translation when ENTERing programs with:

ENTER "filename/a+".

The LIST command stored the program on disk as a true ASCII file, ready to be ENTERed into another version of COMAL.

I won't go into detail about how to get a file from the disk of one system to that of another, but the possible methods include transfer via null modem, direct modem to modem connection, using a BBS or network as an intermediary, or

running a program that will convert or read files between disk formats. [*On the Amiga you also can run a C64 emulator, such as A64, which can be downloaded from PLink. Hopefully more on this next issue.*] Once you get the program to a disk on the destination computer, the rest is easy. Simply type:

ENTER "program name"

and the file will be read in from disk, just as if you were typing it in. If there are any commands in the original program that COMAL doesn't know, it will pause ENTERing the program, place the cursor on the offending word, display an error message, wait for you to either correct it or "rem out" the line by preceding it with "//", and then continue ENTERing the rest of the program. If you used only Common COMAL keywords in the original program (i.e. no package commands or system-specific keywords) it should run on the new machine with virtually no changes!

Here we may come to a little problem. Depending on what computers you are transferring the program between, you might have some difficulty ENTERing it into the new system. This is because of differences in the machines, rather than in COMAL itself. One common problem I've run into is with delimiters. When a program is LISTed to disk, C64 COMAL puts a carriage return (CHR\$(13)) on the end of each line. However, the Amiga uses line feeds (CHR\$(10)) as its delimiter. IBM uses a carriage return followed by a line feed. COMAL generally follows the rules of the host machine in matters such as delimiters and file names, so when you try to ENTER a C64 COMAL program on an Amiga, it expects to see a line feed after each line and can't find any. Therefore it thinks the entire program is one line.

This is a disadvantage when transferring programs between systems. As far as I know COMAL doesn't provide a way to solve it directly. Fortunately, it is a simple matter to write a short program of about a half dozen lines to read in the source file and replace the original delimiters with the ones expected by the host computer. Then you can ENTER it with no problem. [*Also see the CONVERTER program on page 37.*]

I've now had quite a bit of experience passing programs between the Commodore 64, Amiga, and IBM (no, I don't have an IBM; those programs have come from other people) via PLink, and I've learned that COMAL is indeed VERY compatible, especially if you program mostly with keywords that are part of the Common COMAL definition. But even the system-specific keywords can often be simulated on another computer with a PROC, FUNC, or package command.

COMAL Type Detector

by Len Lindsay

Have you ever been curious? Maybe your COMAL program is too. Several of us sent messages back and forth on PLink about these two topics (*part of our message log is at the end of this article*):

How can a program tell what computer it is running on?

How can a program tell what COMAL it is running under?

If a program could identify the computer type, it could then do special things (like reverse video) that are done differently on each computer.

There were two different trains of thought on this. Joel Rea (PLink ID is COMALTODAY) had an interesting method. It assigns an integer to each computer type. Then each version of COMAL running on that computer had a digit following the decimal point (ie, 1.2 and 1.3, where 1 is the computer type, and 2 and 3 are two versions of COMAL running on computer type 1).

My idea was to assign a number to each COMAL that would be part of this "system". A standard set of procedures and functions could be tacked onto any COMAL program. I put together one FUNC and two PROCs:

```
FUNC computer'comal  
PROC computer'dims  
PROC computer'setup
```

To be consistent, I started each name with computer'. This is not necessary, of course, but helps make it obvious that they go together. I also start most of my "global" variable names with computer' as well, for the same reason. I won't list the FUNC and PROCs here, since they are already listed at the very end of the CONVERTER program in this issue.

I do not close the FUNC and PROCs purposely. That way the variables I use are accessible to your program. The variables are named in such a way that they should not conflict with any variable names you may already have used!

COMPUTER'COMAL returns a number that identifies what COMAL is running:

- 1 is Power Driver
- 2 is C64 Cartridge
- 0 is unknown COMAL
- 1 is AmigaCOMAL 2.0
- 2 is CP/M 2.1
- 3 is IBM 2.2 (and 3.0 for now)

4 will be IBM 3.0

I use RANDOMIZE to determine what COMAL is running, since they each have their own way to force a specified set of pseudo random numbers. However, both IBM PC COMALs give the same set of numbers, so I have to come up with another test to tell them apart!

COMPUTER'COMAL is the only thing you need to allow any of your programs to identify what COMAL they are running under. Note that the function also sets up some variables that you might wish to use in your program as well:

```
powerdriver=-1  
c64cart=-2  
unknown'computer=0  
amiga2=1  
cpm2=2  
ibm2=3  
ibm3=ibm2 // for now, later =4
```

It is easy to have your program find out what COMAL it is running under. Like this:

this'computer=computer'comal

Later in the program you could have this:

IF this'computer=amiga2 THEN ...

Or, to check if the computer is a C64/C128:

IF this'computer<0 THEN ...

I also wrote two PROCs that setup some common things that vary between computer types:

```
COMPUTER'SETUP  
COMPUTER'DIMS
```

Continued...

COMAL Type Detector

...Continued

To use them, just have this line at the start of your program:

0010 computer'dims; computer'setup

After that line is executed, you can use these variables:

COMPUTER'DRIVE\$(x)

This assigns six drives for you. You should be able to use drives 0 and 1 safely on most systems. For example, if the program was running on a C64:

**COMPUTER'DRIVE\$(0) ==> "0:"
COMPUTER'DRIVE\$(1) ==> "1:"**

If the program was running on an Amiga:

**COMPUTER'DRIVE\$(0) ==> "DF0:"
COMPUTER'DRIVE\$(1) ==> "DF1:"**

If the program was running on an IBM:

**COMPUTER'DRIVE\$(0) ==> "A:"
COMPUTER'DRIVE\$(1) ==> "B:"**

Thus, even though each computer system has a different way of naming each disk drive, your program could handle it like this:

DIR computer'drive\$(0)

or

This is part of the dialog from PLINK discussing how to implement an automatic way for programs to detect what computer they are running on:

Date: 9/10/90 22:06 Num:1,514
To : DAVID*W (r) By :CAPTAIN C
Title: R#1503 VALUES

I changed the values returned so that C64/128 computers were always less than 0 and IBM's were always greater than or equal 1 (I think). CP/M fits in the 0 spot, since an unknown can be given CP/M values as a good start. Remember, users do *NOT* have to even know what values are being returned if they use the variables that are setup...

IF computer'comal=powerdriver THEN

**PRINT "Place data disk into drive";
PRINT computer'drive\$(0)**

COMPUTER'PRINT\$ & COMPUTER'SCREEN\$

Unfortunately, IBM PC COMAL uses the IBM PC names for its printer and screen, instead of the COMAL standard of "LP:" and "DS:". These two variables will be set properly for whatever COMAL your program is running under. So in your program, to select the printer you could just use this:

SELECT computer'print\$

COMPUTER'RVSON\$ & COMPUTER'RVSOFF\$

One nice touch is available with most computers ... the ability to print in reverse video. However, the way to do it differs on each type of computer. With these two variables you can turn on and off reverse video in your print statements. See the CONVERTER program listed on page 40 to see how they are used.

COMPUTER'EOL\$

Every computer has a way of marking the end of line. Unfortunately, this varies between computers. An Amiga uses CHR\$(10) (linefeed). C64 uses CHR\$(13) (carriage return). IBM PC uses both: carriage return and linefeed. COMPUTER'SETUP assigns the proper EOL character(s) to this variable.

Date: 9/11/90 17:52 Num:1,521
To : CAPTAIN C (r) By :DAVID*W
Title: R#1514 CHECK THE LIBRARY

The reason I thought of distinguishing between CP/M and "unknown" is that the program can test for an unknown computer...

**IF computer'comal=FALSE THEN
or
IF computer'comal=other THEN**

Using "other", the user still would not need to know the actual value returned, but can call an exception routine or something if "other" is returned. Zero just seemed to be the most natural value.

That was one of the changes I suggested in the new COMPUTER'COMAL FUNC that I will be uploading.

Date: 9/11/90 20:40 Num:1,532
To : DAVID*W (r) By :CAPTAIN C
Title: R#1521 MORE COMMENTS NEEDED

Now is the time to fool with the COMPUTER'COMAL function ... before it gets in print. I have to see what you suggested.

I like having all the C64's less than 0. I like having all the IBM's greater than something. The rest can be in between. Maybe unknown could be 0 and CP/M 1 and Amiga 2 and reserve 3 and 4 for future use. Don't want to reserve too many blanks in case people want to have a string array of COMPUTER names for each one, and have

Continued... **Evolution of COMAL Type Detector** ...Continued

to allocate in a DIM for the entire range (-2:5).

Date: 9/11/90 22:16 Num:1,536
To : CAPTAIN C (r) By :DAVID*W
Title: R#1532 TYPES TOGETHER

Keeping the computer types together is a good idea. I like having zero be "unknown". The other computers, maybe including Apple as well, could be in between.

Date: 9/10/90 19:23 Num:1,505
To : CAPTAIN C (r) By :DAVID*W
Title: R#1500 VARIABLE NAMES

Thanks. I'll download the program and look at it. Now, about your comment:

> The SETUP uses a variable named > COMPUTER'RVSONS. Don't change the > SETUP procedure, rather do this in > your program:
> COMPUTER' SETUP
> REVONS:=COMPUTER'RVSONS

How about passing the variables into the SETUP and other PROCedures as parameters? Then you could use any variable name in the outside program. No, you couldn't do that because they have to be DIM'ed before they can be referenced. Or rather, they can't be DIM'ed AFTER they've been referenced. It seems like there should be some way to avoid REVONS:=COMPUTER'RVSONS assignment in my program. Maybe not.

Date: 9/10/90 22:02 Num:1,512
To : DAVID*W (r) By :CAPTAIN C
Title: R#1505 NO PARAMETERS

NO! I don't like the idea of passing parameters into SETUP. That way you are FORCED to use the variables! As it is, they are GLOBAL and you can use them if you want, or just ignore them! Much more friendly that way. Kind of like COMAL. You don't have to use packages, but they are there if you want to try them. You don't have to use POINTERS, but AmigaCOMAL allows you to if you want to.

Date: 9/11/90 18:00 Num:1,523
To : CAPTAIN C (r) By :DAVID*W
Title: R#1512 RE-DIM'ING

There is no way it could be done anyway. If a variable was passed into a PROC as a parameter it could not then be DIM'ed inside the PROC, as I realized after I started typing that message.

Date: 9/14/90 5:28 Num:1,550
To : DAVID*W (r) By :COMALTODAY
Title: R#1536 COMPUTER'COMAL

I still think the best way to do it is to have the INT(computer'comal) be the computer type, and

frac(computer'comal) the COMAL version within that computer type. frac() is a user defined real FUNC that subtracts INT(arg) from its argument, thus returning the fractional part of the argument.

Perhaps negatives could be used for all versions which aren't at least 90% compatible with the Common COMAL standard. With this in mind:

-64.14 = C64 COMAL 0.14
-64.15 = C64 COMAL Power Driver
64.2 = C64 COMAL 2.0 Cart.
64.21 = C64 COMAL 2.0 Cart.
With SuperChip
128.2 = C128 COMAL 2.2 Cart.
88.2 = IBM UniCOMAL 2.2
88.3 = IBM UniCOMAL 3.0
68.2 = AmigaCOMAL 2.0+
etc. etc. etc.

Date: 9/15/90 0:35 Num:1,557
To : COMALTODAY (r) By :CAPTAIN C
Title: R#1550 DEFINITE MAYBE

Joel, that's fine with me *IF* it also allows users to utilize the FUNC in the same manner I have thus far demonstrated. For EXAMPLE:

IF COMPUTER'COMAL=AMIGA2 THEN...
or
IF COMPUTER'COMAL<0 THEN
PRINT "a 40 column screen here"
ENDIF

That is simple and straight forward. No need for an extra step to pull out a value needed, which might be just enough to cause the routines to not be used.

Can your proposed FUNC be used simply to identify 40 col screens and/or PET-ASCII?

Date: 9/17/90 23:16 Num:1,576
To : CAPTAIN C (r) By :COMALTODAY
Title: R#1557 SURE IT CAN!

Sure! All you have to do is assign the constants to names! The idea is to also allow checking solely for computer type without regard to COMAL version when doing hardware-specific stuff like POKEs if needed.

IF INT(computer'type)=64 THEN POKE \$D011, PEEK(\$D011) BITAND \$DF BITOR \$40 //double wrap line

The line above turns on Extended Color Text Mode on C64s only!

Date: 9/15/90 18:10 Num:1,566
To : COMALTODAY (r) By :CAPTAIN C
Title: R#1550 STRING ARRAYS

Joel, I am looking into your idea now. One problem I see is that your range of numbers goes from -88 up to

128. If a user needed a string array based on the values of COMPUTERS, that would be too much.

So, lets say we dropped the numbers down to a usable range of 20 or less (integer-wise).

Now, if the user sets up a string array .. say, of the NAME of the computer and its COMAL, how do they do it? With a straight INTEGER system it is very straight forward..

COMPUTERNAME\$(-2)="C64 Cart"
COMPUTERNAME\$(-1)="Power Driver"
COMPUTERNAME\$(0)="Unknown"
etc...

Lets say your system returned:

-1.0 C128 cart
-2.0 C64 Cart
-2.1 Power Driver

It seems that there is no simple home programmer beginning level way to deal with the numbers returned for use in a string array. And that might be one common use ... to have a string array of something for each computer type that might run the program.

I am looking into this ... do you have any ideas about it?

Date: 9/17/90 23:23 Num:1,577
To : CAPTAIN C (r) By :COMALTODAY
Title: R#1566 HERE'S A WAY!

How about this:

types\$=""64""68""88""128""...
computer\$(1):="Commodore 64"
computer\$(2):="Commodore Amiga"
computer\$(3):="IBM PC or clone"
computer\$(4):="Commodore 128"
etc. ...
computer\$(0):="**unknown**"
PRINT "You are using a";
PRINT computer\$(CHR\$(ABS(
computer'type#))IM types\$)//wrap

Date: 9/18/90 3:15 Num:1,580
To : COMALTODAY (r) By :CAPTAIN C
Title: R#1577 BETTER

If we did that ... then perhaps the TYPE\$ you use should be set in SETUP as COMPUTER'TYPE\$... and use CHR\$ instead of "" markings. It is sounding better. Need other people to comment on it.

Date: 9/18/90 18:51 Num:1,581
To : CAPTAIN C (r) By :DAVID*W
Title: R#1580 COMPUTERS AND COMALS

If changing the COMPUTER'COMAL routines to distinguish computer type as well as COMAL version doesn't add too many lines it would be good.

Continued...

Evolution of COMAL Type Detector

...Continued

We also need a way to distinguish between two different versions of COMAL on the same computer. The way it stands now each different version of COMAL determined by COMPUTER'COMAL *IS* a different computer, except for C64 COMAL 2.0 and Power Driver.

Date: 9/14/90 17:38 Num:1,553
To : CAPTAIN C (r) By :DAVID*W
Title: PROPOSITION FOR CHANGE

I tested the COMPUTER'COMAL routines in C64 COMAL 2.0 and Power Driver. They worked perfectly! So it has now been tested and verified on IBM, Amiga, and C64/C128.

One more proposal on the COMPUTER'COMAL routines...

I suggest that we swap the values returned by amiga2 and amiga3 with cpm2. The actual value that is returned from the FUNC is not important in itself, but by doing it this way the systems that use the CR/LF delimiter will be grouped together.

A program can test for such a condition with a line like:

IF computer'comal>=cpm2 THEN

I just realized you could also accomplish the same thing by saying:

IF computer'eol\$=CHR\$(13)+CHR\$(10)

But the first way looks better and is shorter, and there may be other situations in which it would be advantageous to have CP/M and IBM together.

Is there any system that uses TWO bytes to indicate reverse field, or FIVE bytes as the screen or printer name? Amiga and I think IBM use 3 or 4 for the printer. Oh well, it's just a couple of bytes anyway, and it allows for future expansion.

Date: 9/15/90 0:22 Num:1,556
To : DAVID*W (r) By :CAPTAIN C
Title: R#1553 OK

First, switching AMIGA and CP/M in the values is no big deal. Originally I placed AMIGA with higher values, since it was closer to the new IBM COMAL with POINTERS and STRUCTURES. However, that is not really more than an arbitrary way I had of assigning the values... since the values themselves do not mean anything any way... except if you try to group them as > or as < a certain value. Less Than Zero gives you the C64/128 with PET-ASCII and 40 column screens. Greater Than or Equal to IBM2 gives all the IBM's.

FEEDBACK needed ... would there be times we might want to group them as AMIGA and IBM? Or is this foolish, because we can use the individual values if we want to.

Next, IBM has "LPT1:" which is 5 chars. What about the future when it might be up to "LPT10:"?

Next, IBM uses two bytes for some things, kind of like escape codes. Was allowing for that too.

Both could be dropped, especially since it should be years before we get new COMALs that probably won't need the extra anyway. And if they did, we could adapt to it then. However, I was thinking that planning ahead would be best; cost is 1 byte for PRINTER, 1 byte for SCREEN, and 1 char per each computer type... wait, that is wrong ... cost is 1 blank spot saved for AMIGACOMAL 3 (regardless of whether it is before or after CP/M).

DAVID ... exactly what did you change in the PROCs you posted here? (So I can make changes in mine without having to pull yours out from here.) Just moved CPM2 and the AMIGAS in the order assigned number values?

Date: 9/15/90 11:21 Num:1,560
To : CAPTAIN C (r) By :DAVID*W
Title: R#1556 VALUE CHANGES

> FEEDBACK needed ... would there be times we might want to group them as AMIGA and IBM? Or is this foolish, because we can use the individual values if we want to.

It probably doesn't matter how we do it. There could be a time when you want to group by delimiter and others when you want to group by something else. At any rate, you can individually specify which computer types you want to test for. Of course, this will become more cumbersome as the list increases.

> Next, IBM has "LPT1:" which is 5

Maybe you should leave room for 6.

> DAVID, exactly what did you change in the PROCs you posted here?

I only changed the one line:

```
9040 //cpm2:=1; apple2:=2; amiga2:=3;
      amiga3:=3 //wrap line
to:
9045 amiga2:=1; amiga3:=1; apple2:=2;
      cpm2:=3 //wrap line
```

Date: 9/15/90 13:57 Num:1,561
To : DAVID*W (r) By :CAPTAIN C
Title: R#1560 TWO FOR AMIGA

David, you want AMIGA2 and AMIGA3 to have the same value? Would it be better to have AMIGA2=1 and AMIGA3=2 and APPLE2=3, CPM2=4? Hmmm wait ... I had both as =3 before?? Do you think we should reserve a spot for the future AMIGA COMAL update to 3.0?

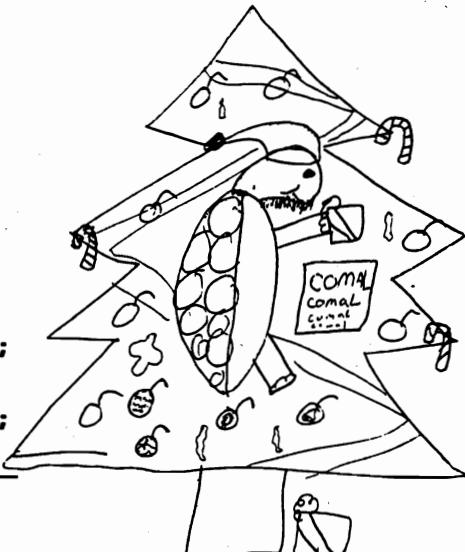
Date: 9/15/90 16:23 Num:1,562
To : CAPTAIN C (r) By :DAVID*W
Title: R#1561 DIFFERENT VALUES

Yes, that's a good idea to give amiga2 and amiga3 different values. I just left them the way they were.

Date: 9/27/90 21:24 Num:1,593
To : CAPTAIN C (r) By :COMALTODAY
Title: R#1586 HERE'S ONE WAY...

How about THIS logic (pseudo-COMAL)?

```
IF INT(computer'comal)=ibm THEN
  INPUT "Drive letter? ": drive$
  INPUT "Path? ": path$
  INPUT "filename.ext? ": name$
  IF ("." IN name$)=0 THEN
    name$+=".TXT"
  ENDIF
  full$=drive$+":"+path$+"\\"+name$
  OPEN FILE 10,full$,READ
ELIF INT(computer'comal)=c64 OR
  INT(computer'comal)=c128 THEN//wrap
REPEAT
  INPUT AT 0,0,2:"Drive num?":drive
  UNTIL drive>7 AND drive<16
  INPUT AT 0,0,16: "Filename?": name$
  full$=str$(drive)+"\\"+name$
  OPEN FILE 10,full$,READ
ELIF INT(computer'comal)=amiga THEN
  INPUT AT 0,0,8: "Device name?": dev$
  INPUT AT 0,0,100: "Path? ": path$
  INPUT AT 0,0,64: "filename?": name$
  full$=dev$+":"+path$+"\\"+name$
  OPEN FILE 10,full$,READ
ELSE
  INPUT AT 0,0,240: "Complete
    filespec? ": filespec$/"\wrap
  OPEN FILE 10,filespec$,READ
ENDIF
```



Converter

by David Warman

This COMAL 2.0 program runs **unaltered** on the Amiga, IBM, and Commodore 64/128 (see below), and should run under most CP/M or other implementations of COMAL as well. There is a separate version for Power Driver included on the C64 Today Disk 26. [*Power Driver does not allow user defined string functions.*]

The primary purpose of this program is as an aid in converting LISTed COMAL programs and text files between different computer systems, but it also has other uses. CONVERTER can be ENTERed into COMAL exactly as listed. It will automatically detect what type of computer it is running on! It will make an educated guess as to what computer the file to be converted came from. It uses that information to automagically set up all conversion options, assuming that you want to convert the file for use on your own system. However, you can override any of the default settings in the event that you want to convert the file for another system or for some special purpose (*such as uploading to a network, like PLink*).

CONVERTER will automatically choose the default settings for Commodore 64 and 128, Amiga, IBM, and probably CP/M systems (*it has not actually been tested in CP/M COMAL*). It should be able to handle files from other computers as well, but may not be able to automatically determine the type of computer.

This program started out as a simple tool to convert between ASCII and PetASCII, and change the delimiters used on various systems, but it has since evolved into a much more sophisticated and easy-to-use program.

HISTORY

I got the idea to write CONVERTER some time after I bought an Amiga and started transferring C64 COMAL programs to the Amiga. I learned that you can't simply LIST a program in C64 COMAL and then ENTER it into AmigaCOMAL directly. Most of the problems lie in the differences in the computer systems rather than in COMAL itself. For instance, Commodore 8-bit machines use a carriage

return (CR) as the delimiter in files. This applies to LISTed COMAL programs as well as most word processor text files and data files. When you LIST a program to disk, each line is stored as PetASCII text and a CR is added to indicate the end of the line.

But not all computers use CR's as delimiters. The Amiga uses a line feed (LF) and IBM needs both a CR and a LF. So when you try to ENTER a LISTed C64 program into AmigaCOMAL, it looks for a LF to indicate end-of-line and can't find any, so it thinks the entire program is one line! That's where this program was born.

At first it was just a simple utility that I used to convert CR's to LF's and PetASCII to ASCII. Then I decided to upload it to the "Workshop" section in People/Link's COMAL Club and ask for suggestions for improvements. (*The Workshop is an area where you can upload a program that you are working on for help and advice, or testing and suggestions. It is also a good way to work on a group programming project with people throughout the country, as I discovered later.*) I got some suggestions pretty quickly. It was suggested that the program should be able to add line numbers to files that had been DISPLAYed to disk, or renumber lines in LISTed files, so I added that option.

**Using the WorkShop is a good
way to work on a group
programming project with people
all around the country!**

Along the way I learned about XModem padding. People Link uses XModem or the fast WXModem protocol to transfer files, both of which require that the file be a multiple of 128 bytes. If it isn't, the terminal program adds "padding" bytes to the end of the file to bring it up to the next multiple of 128. This is fine for archived files, since most archivers ignore the padding when they extract the individual files. However, XModem padding can be a problem with COMAL Club files, since they are often uploaded in plain LISTed form so that they can be

Converter

downloaded and ENTERed into any version of COMAL. Although the padding usually isn't harmful, COMAL doesn't like seeing padding characters on the end of a file that it is ENTERing, so I added an option to the program to remove the padding added by XModem to prevent any chance of losing the last line.

CONVERTER rested on PLINK while I went on to other projects. After a couple of months I went back to work on it. I improved the user interface by adding a menu, making it much more user-friendly, and added the ability for it to convert straight text files in addition to LISTed COMAL programs. Then I uploaded the new version. This is where things really got busy!

Len Lindsay downloaded it and realized that with just a slight modification it could run unchanged on the C64/128, Amiga, and IBM, and probably most other COMAL's. He made those few changes and re-uploaded it. About this time he was developing a FUNCtion to determine what computer and version of COMAL a program is running under (*it is discussed on page 33*) and uploaded it to PLINK. I was able to incorporate that routine into CONVERTER to enable it to automatically select several settings which previously had to be entered manually.

Now it was easy to use, but Len thought that it could look better. He made some cosmetic changes, especially to the menu. While the program RUNS fine on the C64, the menu and text screens were written for an 80-column display and didn't look good on the 64's 40-column screen. On the other hand, they would look bad on an 80-column screen if I only used 40 columns. Len devised a clever method to display the text on both 40- and 80-column screens, using the full 80 columns and still remaining readable in 40 columns (*based on the September 1990 Challenge - see page 48*).

Now the program was going through a flurry of changes, undergoing as many as 23 revisions in 15 days at one point! Usually one person would make some changes, upload the file, then wait for the next person to take their turn before downloading the next version to make more changes. That's one of the really exciting aspects of a team programming

project on a computer network. We came up with a method of marking our changes so the next person to download it would know where the changes had been made since the last revision. It got to the point where I eagerly anticipated downloading the newest version to see what changes had been made!

If you list the beginning of the program you will see the revisions that it went through (removed from the printed listing, but they are in the program on *Today Disk 26*). Now, on to the details of CONVERTER's operation.

PROGRAM OPTIONS

The first thing you do is enter the name of the file you want to convert and a name for the output file. Just press RETURN to get a listing of the current directory. For the output file, you can hit RETURN to use the default name or overtype it with another one. The output file name must be different than that of the input file. If the output file already exists you will be asked if you want to overwrite it. [*Look at the program listing to see how it creates a suggested output file name for you.*]

CONVERTER then scans the source file looking for certain common delimiters. If it can't find a CR, LF, or CR+LF, it will try to find line numbers, as in a LISTed program, and assumes that the byte immediately preceding the line number is a delimiter. If this also fails, as it might if the file is a text file rather than a LISTed COMAL program, then it will ask the user to enter the ASCII value of the byte(s) used as the delimiter in the source file. You may enter up to three bytes -- I doubt that any system uses delimiters longer than three bytes. If you don't know what the delimiter is, you may be able to find out by examining a dump of the file, or by viewing it with a disk editor. The program must know what the delimiter is before it can proceed further. It is very unlikely that this would be a problem, and should never happen on a file from a Commodore, Amiga, or IBM.

Next you will come to the main options screen. This screen shows all the options available for converting the file. If you are running the program on a C64, Amiga, or IBM, the defaults should already be set and you just need to press RETURN to start the

conversion. You will also be shown a sample of what the file looks like so you can determine whether or not it needs ASCII/PetASCII conversion. This preview sample is necessary because even though the file may have come from a C64, for example, and is being converted for an Amiga, it could have been ASCII-converted with the "/a+" attribute when it was LISTed to disk from C64 COMAL 2.0 and wouldn't need further converting. If the case of the letters in the preview sample looks reversed, then it will need ASCII conversion.

If you're converting the file for use on your own system, the options will be preset for you! Just hit <Enter>.

If you're converting the file for use on your own system, you should be able to just press RETURN without having to change any of the options explained below. If you want to convert the file for some other system, simply select the target system and CONVERTER will set the options appropriate for that system.

System conversion options:

C - Convert file for Commodore 64/128.

A - Convert file for Amiga.

I - Convert file for IBM.

These three options determine what delimiter is used for the output file. The input and output delimiters are displayed in the reversed bars to the right of the file name and target system name, respectively. Selecting any of these options may change the ASCII-conversion flag, but you can still override it with the T option. You can use these options to prepare a file for a system other than your own; they will change the delimiter to that expected by the target computer you choose.

O - Convert the file for some system other than Commodore, Amiga, or IBM. To use this option, you must know what byte(s) the target system uses for its delimiters. The program will ask you to enter the required delimiter.

RETURN - Start the conversion. The output file will be displayed on the screen (and optionally on the printer), in addition to being written to the output file.

STOP/ESC/BREAK etc - Pressing the key that your computer uses as the "stop" or "escape" key will abort the conversion.

Other options and the keys to press to select them are as follows:

P - Printer Select. If this is ON, then the file will be listed to the printer as it is being converted, in addition to being displayed on the screen.

T - Translate between ASCII and PetASCII.

X - XModem padding stripping. When ON it strips XModem padding off the end of the file. The byte value used for XModem padding is different for different computers and terminal programs. CONVERTER considers just about any unprintable character or non-delimiter at the end of a file to be XModem padding. Even if there is no padding, leaving this option ON won't hurt anything.

U - This option will only be displayed if you are converting a file to C64 format. It determines which set of the C64's uppercase characters to use. There is one set with values from 65-90 and another one from 193-218. If you are converting a LISTed program for C64 COMAL 2.0, you must use the second set; the first set will not ENTER correctly. Thus, the default is set 2.

R - Renumbers lines in a LISTed COMAL program, or add line numbers to lines without numbers, such as DISPLAYed programs or lines extracted from a text file. If you toggle this option ON you will be given the additional choice of specifying the starting line number and increment. Pressing "L" will allow you to first enter the starting line number, then the increment to use between line numbers. If you are converting the file for IBM PC COMAL 3.0, you will be given the further option to choose 4- or 5-digit line numbers. If the line numbers exceed the maximum value allowed by the

version of COMAL running on the target computer, the program will abort the conversion and you will have to start over again, using a lower starting number or a smaller increment. Choosing the **Renumber** option will automatically turn off **Strip Line Numbers**.

S - Strip line numbers toggle. This option, like the **Renumber** option, is useful for LISTed COMAL programs only. It strips the line number from each line, making a file that looks like a DISPLAYed file, with indentation intact. This allows manipulating a program file with a word processor. Without the line numbers, you can rearrange the order of the PROCedures and FUNCtions with a word processor, then use the **RENUMBER** option to add the line numbers back! You may not select the **Renumber** option while **Strip Line Numbers** is set.

/ - This option toggles removal of comments from programs. If it is ON, all text after and including "://" will be removed from each line, unless the "://" is found inside quotes.

B - Change blank lines to "://". Toggling this option ON will change all blank lines in a program, such as may be found separating PROCs and FUNCs, to a comment ("://"). Power Driver ignores blank lines in programs being ENTERed, so it is necessary to change all blank lines to comments in order to retain a separator line between program sections. If used simultaneously with the **/** option above, these blank line comments will not be removed; this option has priority.

Commodore users please note: This is for reference only, since the version of CONVERTER on the Today Disk will be ready to run. This program runs unchanged on all systems, except for ONE line on the 64/128. You will notice the flashing box on the options screen; getting the timing of the flashes to be the same on all computers was one of the trickiest things. It was especially difficult since different models of the Amiga and IBM run at different clock speeds. We finally settled on the TIME/TIMER method used in the PROCedure **computer'tenths**. On Commodore, the TIME line should be used, while the TIMER line is used on

Amiga and IBM. If you run the program on a different computer, make sure the correct line is "commented out".

80-column users will notice some extras that don't show up on a 40-column screen. These are mainly simple explanations that appear when you select various options. However, it is all explained in this article, so 40-column users aren't left out.

Future plans for this program include the ability to translate the extended characters (ASCII 128 and up) between computer systems where possible.

```
// File CONVERTER v2.0 - for Amiga, IBM, C64/128, CP/M
// By David Warman, October 5, 1990
// C64 users: swap // lines in PROC "computer'tenths"
//
program
//
PROC program
init
REPEAT
  linenum$=""; line'number:=default'line
  escape:=false //init
  process
  PRINT
  PRINT revon$;"Convert another file?";revoff$;
UNTIL yes'no$ IN "nN"
page
PRINT AT 3,1: "Converter Concluded!"
TRAP ESC+
ENDPROC program
//
PROC init
  computer'dims;computer'setup //system setup
  chunk'size:=2000 //bytes to get from file
  DIM infile$ OF 80 //input filename
  DIM outfile$ OF 80 //output filename
  DIM newfile$ OF 80 //temp storage for Amiga
  DIM chunk$ OF chunk'size //input characters
  DIM reply$ OF 1 //1-character replies
  //delimiters
  DIM old'delim$ OF 3, new'delim$ OF 3, delim$ OF 3
  DIM temp$ OF 3 //work variable
  DIM line$ OF chunk'size //cur line being translated
  DIM pad$ OF 140 //store chars that could be padding
  //line search, current line # for renum
  DIM num$ OF 5, linenum$ OF 6
  DIM revon$ OF 1, revoff$ OF 1 //reverse on and off
  DIM preview'chunk$ OF 600 //temp storage prev sample
  DIM zeroes$ OF 4, last'char$ OF 1
  zeroes$=="0000" //prepend to short line numbers
  pad$="" //initialize possible pad characters
  FOR j:=0 TO 9 DO pad$:=chr$(j) //build up string with
  FOR j:=11 TO 12 DO pad$:=chr$(j) //possible pad
  FOR j:=14 TO 31 DO pad$:=chr$(j) //characters
  FOR j:=127 TO 192 DO pad$:=chr$(j)
  FOR j:=219 TO 255 DO pad$:=chr$(j)
  add'numbers:=false; increment:=10; digit:=4
  strip'numbers:=false; strip'pad:=true
  remove'comments:=false; no'blanks:=false
  //from COMPUTER'SETUP
  revon$:=computer'rson$; revoff$:=computer'rvsoff$
  new'delim$:=computer'eol$
  //automatically detect computer type
```

Continued...

Converter

...Continued

```
computer:=computer'comal
target:=computer //default target computer
other:=unknown'computer //not determined
printer:=false //no echo lines on printer
upperset:=2; maxline:=80
IF computer<0 THEN maxline:=40 //C64
center:=20 //offset to center a line
endtab:=40 //spec text output (41 on C64 = next line)
IF computer<0 THEN endtab:=41; center:=0 //c64
default'line:=10
ENDPROC init
//
PROC process
get'info
TRAP ESC-
IF file'exists(infile$) THEN
page
find'endings
needs'converting:=false //don't convert ASCII
IF target=c64cart AND old'delim$<>chr$(13) THEN
needs'converting:=true //wrap line
IF old'delim$=chr$(13) AND target<>c64cart THEN
needs'converting:=true //wrap line
preview
options
page
TRAP
    IF NOT (escape OR esc) THEN convert'file
    IF printer THEN eject'page
HANDLER // come here if max line # exceeded
abort
ENDTRAP
ELSE
    PRINT AT 15,1: revon$;"File not found.";revoff$
ENDIF
ENDPROC process
//
PROC abort
//make sure printer is off
IF printer THEN SELECT OUTPUT computer'screen$
CLOSE
PRINT revon$;"Aborting conversion!";revoff$
IF err=555 THEN //max line # exceeded
    PRINT "Max line number exceeded!"
    PRINT "Delete the incomplete file?";
    reply$:=yes\no$
    IF reply$ IN "yY" THEN DELETE outfile$
ELIF err=999 THEN
    PRINT "Aborted by user!!"
ELSE
    PRINT errtext$
ENDIF
ENDPROC abort
//
PROC eject'page
SELECT OUTPUT computer'print$
page
SELECT OUTPUT computer'screen$
ENDPROC eject'page
//
PROC get'info
REPEAT
get'infile
get'outfile
check'outfile
UNTIL infile$<>outfile$ AND reply$ IN "yY"
ENDPROC get'info
//
PROC get'infile
REPEAT
title'page
PRINT AT 5,1: "File to convert: (<ENTER> for DIR)"
```

```
INPUT AT 6,1,maxline: """: infile$
IF infile$="" THEN show'dir
UNTIL infile$>"" // must have some name
PRINT AT 15,1: spc$(maxline) //erase message if one
IF NOT (file'exists(infile$)) THEN
    PRINT AT 15,1: revon$;"File Not Found.";revoff$
    infile$="" //reset to null if not found
    waitkey
ENDIF
UNTIL infile$>"""
ENDPROC get'infile
//
PROC get'outfile
REPEAT
IF "." IN infile$ THEN //check if extension is used
    // change it to .new
    outfile$:=infile$(1:"." IN infile$)+"new"
ELSE // no extension yet ... just add .new
    outfile$:=infile$+".new"
ENDIF
PRINT AT 8,1: "Output filename: (Must be different)"
PRINT outfile$,TAB(maxline)
newfile$:=outfile$
INPUT AT 9,1,maxline: """: outfile$
IF outfile$="" THEN outfile$:=newfile$ //for Amiga
IF outfile$=infile$ THEN PRINT AT 15,1:
    revon$;"Output name must be different";revoff$/wrap
UNTIL outfile$>infile$
PRINT AT 15,1: TAB(maxline) //erase line
ENDPROC get'outfile
//
PROC check'outfile
reply$=="y" //init
IF file'exists(outfile$) THEN
    PRINT AT 15,1: revon$;"FILE EXISTS! - Overwrite
it?";revoff$/wrap line
    reply$:=yes\no$
    PRINT AT 15,1: TAB(maxline) // erase error message
    //Amiga can't read from screen
    IF computer=amiga2 THEN page
ENDIF
ENDPROC check'outfile
//
PROC popup
popcol:=7+center; poprow:=3 // row and column of popup
PRINT AT poprow,popcol: revon$;"Change options by
pressing";revoff$/wrap line
PRINT AT 0,popcol: revon$;"the indicated key
below...";revoff$/wrap line
PRINT AT 0,popcol: revon$;"..... ** OR ** ....."
....";revoff$/wrap line
PRINT AT 0,popcol: revon$;"Press <ENTER> to start
the";revoff$/wrap line
PRINT AT 0,popcol: revon$;"conversion....."
.....";revoff$/wrap line
ENDPROC popup
//
PROC options
clear'keyboard
menu
REPEAT
    preview'sample
    pausekey(10)
    IF reply$>chr$(0) THEN choice
    IF NOT (escape OR reply$=chr$(13)) THEN
        popup
        pausekey(20)
        IF reply$>chr$(0) THEN choice
    ENDIF
    UNTIL reply$=chr$(13) OR escape
ENDPROC options
//
PROC menu
```

Continued...

Converter

...Continued

```
target'line
target'menu
upper'menu
printer'menu
PRINT " T=Translate ASCII.....",
y'or'n$(needs'converting); //wrap line
IF computer>=0 AND target<0 THEN PRINT TAB(39),"<=
convert it to PET-ASCII", //wrap line
PRINT TAB(maxline) //end of line
PRINT " /=Remove // comments.....",
y'or'n$(remove'comments), //wrap line
IF computer>=0 AND remove'comments THEN
    PRINT TAB(39),"<= remove ""//"" and rest of line",
ENDIF
PRINT TAB(maxline) //end of line
PRINT " X=Strip XModem padding.....",
y'or'n$(strip'pad) //wrap line
PRINT " S=Strip Line numbers.....",
y'or'n$(strip'numbers), //wrap line
IF computer>=0 AND strip'numbers THEN PRINT TAB(39),
"<= remove line numbers", //wrap line
PRINT TAB(maxline) //end of line
blank'menu
renumber'menu
ENDPROC menu
//
PROC blank'menu
PRINT " B=Blank Lines changed to //..",
y'or'n$(no'blanks), //wrap line
IF computer>=0 AND no'blanks THEN
    PRINT TAB(39),"<= replace blank lines with ""//"",
ENDIF
PRINT TAB(maxline) //end of line
ENDPROC blank'menu
//
PROC target'line
PRINT AT 9,1: revon$,"Target ==>";
IF target<0 THEN PRINT "Commodore 64/128";
IF target=ibm2 THEN PRINT "IBM PC & PS/2";
IF target=amiga2 THEN PRINT "Amiga";
IF target=cpm2 THEN PRINT "CP/M";
IF target=other THEN PRINT "OTHER";
over:=80-20-len(delimiter$(new'delim$))
IF computer>=0 THEN PRINT TAB(over),"DELIMITER OUT
==>; //wrap line
PRINT delimiter$(new'delim$),TAB(maxline);revoff$
ENDPROC target'line
//
PROC renumber'menu
PRINT " R=Renumber/Add line numbers..",
y'or'n$(add'numbers) //wrap line
IF add'numbers THEN
    show'linenumber
ELSE //blank out area
    PRINT TAB(maxline)
    PRINT TAB(maxline)
ENDIF
ENDPROC renumber'menu
//
PROC printer'menu
PRINT AT 11,1: " P=Printer output
("computer'print$,""), //wrap line
FOR dots:=1 TO 10-len(computer'print$) DO PRINT ".",
PRINT " ",y'or'n$(printer);
IF computer>=0 THEN //80 col only
    PRINT TAB(39),"<=";
    IF printer THEN
        PRINT "also prints on printer as converted";
    ELSE
        PRINT "prints only on screen as converted";
    ENDIF
ENDIF
PRINT //end of line

ENDPROC printer'menu
//
ENDPROC printer'menu
//
PROC upper'menu
//clear line in case switching from C64 out
PRINT AT 10,1: spc$(maxline)
IF target<0 THEN
    PRINT AT 10,1: " U=Upper case set (for C64)...";
upperset, //wrap line
IF computer>=0 THEN //80 col only
    PRINT TAB(39),"<=";
    IF upperset=1 THEN
        PRINT "normal uppercase for word processors",
    ELSE
        PRINT "uppercase for C64 COMAL 2.0",
            TAB(maxline), //wrap line
    ENDIF
ENDIF
PRINT //end of line
ENDIF
ENDPROC upper'menu
//
PROC target'menu
PRINT AT 20,9+center: revon$;"Select Target System:";
revoff$ //wrap line
PRINT AT 0,9+center: revon$;"C=Commodore A=Amiga";
revoff$ //wrap line
PRINT AT 0,9+center: revon$;"I=IBM O=Other";
revoff$ //wrap line
ENDPROC target'menu
//
PROC show'linenumber
PRINT USING " L=Line: ##### Increment: #####";
line'number,increment; //wrap line
IF computer>=0 THEN PRINT TAB(39),"<= Starting line
number & increment", //wrap line
PRINT //end of line
IF target=ibm3 THEN
    PRINT " 5=5 digit line numbers? ....";
    y'or'n$(digit=5), //wrap line
    IF digit=4 THEN
        PRINT TAB(39),"<= use 4 digit line numbers",
    ELSE
        PRINT TAB(maxline),
    ENDIF
    PRINT //end of line
ELSE
    PRINT TAB(maxline),
ENDIF
PRINT //end of line
ENDPROC show'linenumber
//
PROC choice
CASE reply$ OF
WHEN "a","A" // Amiga
    new'delim$:=chr$(10)
    target:=amiga2
    needs'converting:=(source=c64cart)
WHEN "c","C" // Commodore 64/128
    new'delim$:=chr$(13)
    target:=c64cart
    needs'converting:=(source<>c64cart)
WHEN "i","I" // IBM
    new'delim$:=chr$(13)+chr$(10)
    target:=ibm2
    needs'converting:=(source=c64cart)
WHEN "o","O"
    other'system
    target:=other
    needs'converting:=(source=c64cart)
WHEN "t","T"
    needs'converting:=NOT needs'converting
WHEN "x","X"
    strip'pad:=NOT strip'pad
```

Converter

```

WHEN "/"
    remove'comments:=NOT remove'comments
WHEN "r","R"
    add'numbers:=NOT add'numbers
//can't have both
IF add'numbers THEN strip'numbers:=false
WHEN "l","L"
    add'numbers:=true //make sure this is ON
    menu // make sure it is displayed too
    get'linenumber
    default'line:=line'number
WHEN "u","U"
    upperset:=3-upperset // 1 or 2
WHEN "p","P"
    printer:=NOT printer
WHEN chr$(27),chr$(3),chr$(164)//ESC 164=C64 leftarrow
    escape:=true
    RETURN
WHEN "%"
    digit:=9-digit //4 or 5
    add'numbers:=true
    strip'numbers:=false //can't have both
    IF digit=4 AND line'number>9999 THEN force4
WHEN "4"
    digit:=4
    IF line'number>9999 THEN force4
WHEN "s","S"
    strip'numbers:=NOT strip'numbers
    add'numbers:=false
WHEN "b","B"
    no'blanks:=NOT no'blanks
OTHERWISE
    NULL
ENDCASE
//only ibm 3.0 allows 5 digit lines
IF target<>ibm3 THEN only4
    menu //reprint menu with changes
ENDPROC choice
//
PROC force4
    line'number:=10 //forced to reset
    menu //redisplay menu with proper line # field
    get'linenumber
ENDPROC force4
//
PROC only4
    digit:=4
    IF line'number>9999 THEN force4
ENDPROC only4
//
PROC get'linenumber
    INPUT AT 18,10+(5-digit),digit: ":"; num$
    IF valid(num$) THEN line'number:=val(num$)
    IF line'number<1 THEN line'number:=1
    INPUT AT 18,28,4: ":"; num$
    IF valid(num$) THEN increment:=val(num$)
    IF increment<1 THEN increment:=1
    add'numbers:=true
ENDPROC get'linenumber
//
PROC other'system
    page
    PRINT "Enter the number(s) the target computer";
    PRINT "uses as delimiters. These usually are:"
    PRINT " 13 (Carriage Return)"
    PRINT " 10 (Line Feed)"
    PRINT "or a combination of them."
    PRINT
    new'delim$:=get'delim$
ENDPROC other'system
//
PROC find'endings
    PRINT "Working..."
OPEN FILE 2,infile$,READ
chunk$:=get$(2,chunk'size)
CLOSE FILE 2
IF chr$(13)+chr$(10) IN chunk$ THEN //probably IBM
    old'delim$:=chr$(13)+chr$(10)
    source:=ibm2
ELIF chr$(13) IN chunk$ THEN //probably C64
    old'delim$:=chr$(13)
    source:=c64cart
ELIF chr$(10) IN chunk$ THEN //probably Amiga
    old'delim$:=chr$(10)
    source:=amiga2
ELSE //not IBM, Amiga or C64
    last'resort
    source:=cpm2 // ?
ENDIF
old'dsize:=len(old'delim$)
ENDPROC find'endings
//
PROC last'resort
    delim'found:=false
    FOR j:=10 TO 90 STEP 10 DO
        num$=="00"+str$(j)+" " // find line #
        pos:=num$ IN chunk$
        IF pos>5 THEN //make sure it's not 1st line
            old'delim$:=chunk$(pos-1:pos-1)//byte before line#
            delim'found:=true
        ENDIF
    ENDFOR j
    IF NOT delim'found THEN
        delim'message1
        old'delim$:=get'delim$
    ENDIF
ENDPROC last'resort
//
FUNC get'delim$
    delim'message2
    delim$="" //init
    REPEAT
        INPUT AT 0,0,3: "Delimiter: ": temp$
        IF valid(temp$) THEN delim$:=chr$(val(temp$))
    UNTIL temp$=""
    RETURN delim$
ENDFUNC get'delim$
//
PROC convert'file
    OPEN FILE 2,infile$,READ
    DELETE outfile$
    OPEN FILE 3,outfile$,WRITE
    chunk$=""; pad'bytes:=0
    WHILE NOT eof(2) DO
        chunk$+=get$(2,chunk'size-len(chunk$))
        IF eof(2) THEN check'for'padding
        WHILE old'delim$ IN chunk$ DO
            eol:=true
            line$:=extract'line$(chunk$,old'delim$)
            write'line(line$)
        ENDWHILE
        IF len(chunk$)=chunk'size OR (eof(2) AND
            len(chunk$)) THEN //wrap line
            eol:=false
            write'line(chunk$)
            chunk$=""
        ENDIF
    ENDWHILE
    CLOSE
    IF pad'bytes THEN pad'message
ENDPROC convert'file
//
PROC check'for'padding
    ln:=len(chunk$); last'char$:=chunk$(ln:ln)
    IF (last'char$ IN pad$) AND strip'pad=true THEN
        strip'padding(chunk$,last'char$) //wrap line

```

Continued...

Converter

...Continued

```
ENDPROC check'for'padding
//PROC pad'message
PRINT "XModem Padding Removal:";
FOR j:=1 TO pad'bytes DO PRINT ".";
PRINT //end of line
ENDPROC pad'message
//PROC write'line(REF line$)
IF remove'comments THEN delete'comments
IF needs'converting THEN convert'ascii(line$)
IF add'numbers OR strip'numbers OR no'blanks THEN
separate'number //wrap line
PRINT FILE 3: linenum$+line$,
IF eol THEN PRINT FILE 3: new'delim$, //add delimiter
lineout(linenum$+line$,eol)
IF printer THEN
SELECT OUTPUT computer'print$
lineout(linenum$+line$,eol)
SELECT OUTPUT computer'screen$
ENDIF
IF esc THEN REPORT 999
ENDPROC write'line
//PROC delete'comments
pos'rem:="//" IN line$
IF pos'rem=1 THEN
line$="" //entire line is remark, remove it
ELIF pos'rem>1 THEN //remark not 1st character
quote'count:=0 //check odd num quotes before '//'
FOR j:=1 TO pos'rem DO
IF line$(j:j)=chr$(34) THEN quote'count:=NOT
quote'count //wrap line
ENDFOR j
IF NOT quote'count THEN line$:=line$(1:(//" IN
line$)-1) // cut at //' //wrap line
ENDIF
ENDPROC delete'comments
//FUNC blank'line
IF line$="" OR line$=spc$(len(line$)) THEN
RETURN true
ELSE
RETURN false
ENDIF
ENDFUNC blank'line
//PROC lineout(outline$,eol)
PRINT outline$,
IF eol THEN PRINT //end of line
ENDPROC lineout
//PROC separate'number
TRAP //remove # from line
pos:=1; linenum$=""
WHILE line$(pos:pos) IN "0123456789" DO
linenum$+line$(pos:pos); pos:=+1 //wrap line
IF len(linenum$) THEN // if line starts with number
//remove number and any following space
line$:=line$(pos+(line$(pos:pos)=" "):)
ENDIF
HANDLER
line$="" //line # only
ENDTRAP
IF linenum$<>"" THEN linenum$+=" " //tack on space
//Now decide what to do with the line #
IF add'numbers THEN
IF (digit=5 AND line'number>65534) OR (digit=4 AND
line'number>9999) THEN //wrap line
REPORT 555 // force an error - line limit exceeded
ENDIF
linenum$:=str$(line'number) //used if renumbering
IF len(linenum$)<digit THEN//prepend zeroes needed
linenum$:=zeroes$(1:digit-len(linenum$))+linenum$
```

```
ENDIF
linenum$+=" " //add space after line #
line'number:=increment //ready for next line
ELIF strip'numbers THEN
linenum$=""
ENDIF
IF no'blanks AND blank'line THEN line$://""
ENDPROC separate'number
//
PROC preview
preview'message
PRINT
waitkey
page
OPEN FILE 2,infile$,READ
chunk$:=get$(2,1000)
CLOSE FILE 2
preview'chunk$:=chunk$
ENDPROC preview
//
PROC preview'sample
chunk$:=preview'chunk$ //init to repeat preview
lines:=0
preview'line
FOR row:=2 TO 8 DO
IF len(chunk$)>maxline-1 THEN
IF old'delim$ IN chunk$(1:maxline-1) THEN
line$:=extract'line$(chunk$,old'delim$)+
spc$(maxline) //wrap line
ELSE
line$:=chunk$(1:maxline-1)
chunk$:=chunk$(maxline:)
ENDIF
ELSE
line$:=chunk$+spc$(maxline); chunk$=""
ENDIF
PRINT AT row,1: line$(1:maxline-1)
ENDFOR row
ENDPROC preview'sample
//
PROC preview'line
PRINT AT 1,1: revon$,
PRINT "FILE ==>;infile$;
over:=80-19-len(delimiter$(old'delim$))
IF computer>=0 THEN PRINT TAB(over),"DELIMITER IS
==>; //wrap line
PRINT delimiter$(old'delim$),
PRINT TAB(maxline);revoff$
ENDPROC preview'line
//
FUNC extract'line$(REF string$,mark$)
length:=len(mark$)
pos:=mark$ IN string$
IF pos>1 THEN
line$:=string$(:pos-1)
ELSE
line$=""
ENDIF
IF pos+length-1<len(string$) THEN//delim not last byte
string$:=string$(pos+length:)
ELSE
string$=""
ENDIF
RETURN line$
ENDFUNC extract'line$
//
PROC convert'ascii(REF string$)
IF target<>c64cart THEN //CBM to ASCII
FOR i#:1 TO len(string$) DO
byte:=ord(string$(i#:i#))
CASE true OF
WHEN byte>64 AND byte<91 //lower case
```

Continued...

Converter

...Continued

```
byte:+32
WHEN byte>96 AND byte<123 //upper case
  byte:-32
WHEN byte>192 AND byte<219 //upper case
  byte:-128
WHEN byte=164 //back arrow
  byte:=95 //underscore
WHEN byte=92 //Eng. lb.
  byte:=124 //vert. bar
OTHERWISE
  NULL
ENDCASE
string$(i#:i#):=chr$(byte)
ENDFOR i#
ELSE //ASCII to CBM
FOR i#:=1 TO len(string$) DO
  byte:=ord(string$(i#:i#))
CASE true OF
  WHEN byte>64 AND byte<91 //upper case
    IF upperset=2 THEN
      byte:+128 //2nd uppercase set
    ELSE
      byte:+32
    ENDIF
  WHEN byte>96 AND byte<123 //lower case
    byte:-32
  WHEN byte=95 //underscore
    byte:=164 //back arrow
  WHEN byte=124 //vert. bar.
    byte:=221 //shift -
  WHEN byte=96 //back. apos. - approximation
    byte:=39 //apostrophe
  WHEN byte=92 //back. slash -approx.
    byte:=47 //slash
  WHEN byte=123 //lf. brace
    byte:=91 //lf. brack.
  WHEN byte=125 //rt. brace
    byte:=91 //rt. brack.
OTHERWISE
  NULL
ENDCASE
string$(i#:i#):=chr$(byte)
ENDFOR i#
ENDIF
ENDPROC convert'ascii
//PROC strip'padding(REF string$,byte$)
  j:=len(string$); pad'bytes:=0
  WHILE string$(j:j)=byte$ AND j>1 DO
    string$:=string$(1:j-1) // strip last byte
    j:-1; pad'bytes:+1
  ENDWHILE
  //remove last pad byte
  IF string$=byte$ THEN string$="""
ENDPROC strip'padding
//PROC show'dir
page
PRINT "Hit <SPACE> to pause the DIR listing."
waitkey
DIR
waitkey
page
ENDPROC show'dir
//PROC title'page
page
PRINT AT 1,8+center: "Common COMAL Converter"
PRINT TAB(12+center),"by David Warman"
PRINT TAB(6+center),"with help from Len Lindsay"
ENDPROC title'page
//PROC preview'message
page
PRINT "Next you will see a sample of what the",
TAB(endtab), //wrap line
PRINT "file looks like now. If the UPPER and" //end77
PRINT "lower case look reversed, or there are",
TAB(endtab), //wrap line
PRINT "strange symbols... then the file needs" //end77
PRINT "ASCII translation for your system. If",
TAB(endtab), //wrap line
PRINT "you are converting the file for use on" //end77
PRINT "a different system, make sure that you",
TAB(endtab), //wrap line
PRINT "use the correct ASCII for THAT system." //end77
PRINT "I believe that only Commodore 64 / 128",
TAB(endtab), //wrap line
PRINT "systems use a NON-standard ASCII."
PRINT
PRINT "This program automatically sets up all",
TAB(endtab), //wrap line
PRINT "conversion options for *YOUR* system!!" //end77
ENDPROC preview'message
//
PROC delim'message1
page
PRINT "Cannot find a delimiter in the file:"
PRINT infile$
PRINT "The file must come from a system using",
TAB(endtab), //wrap line
PRINT "non-standard delimiters. If possible," //end77
PRINT "enter the number(s) used as delimiters",
TAB(endtab), //wrap line
PRINT "on the system this file came from. If" //end77
PRINT "you don't know, you should be able to",
TAB(endtab), //wrap line
PRINT "find out by examining a hex or decimal" //end77
PRINT "file display. The byte(s) used as the",
TAB(endtab), //wrap line
PRINT "delimiter should be immediately before" //end77
PRINT "each line number in LISTed COMAL files",
TAB(endtab), //wrap line
PRINT "or following each line or paragraph in" //end77
PRINT "text files."
PRINT
ENDPROC delim'message1
//
PROC delim'message2
PRINT "If the original file delimiter is more",
TAB(endtab), //wrap line
PRINT "than one byte, enter the values one at" //end77
PRINT "a time. Press <ENTER> after each one.",
TAB(endtab), //wrap line
PRINT "After entering the last number, press" //end77
PRINT "<ENTER> by itself. Each number must be",
TAB(endtab), //wrap line
PRINT "in the range 0-255, and no more than 3" //end77
PRINT "separate numbers will be accepted."
ENDPROC delim'message2
//
FUNC delimiter$(delim$)
CASE delim$ OF
  WHEN chr$(13)
    RETURN "(CR)"
  WHEN chr$(10)
    RETURN "(LF)"
  WHEN chr$(13)+chr$(10)
    RETURN "(CR+LF)"
  OTHERWISE
    RETURN "(??)"
ENDCASE
ENDFUNC delimiter$
//
FUNC file'exists(filename$) CLOSED
TRAP
```

Continued...

Converter

...Continued

```
f:=7
OPEN FILE f,filename$,READ
CLOSE FILE f
RETURN true
HANDLER
CLOSE FILE f
RETURN false
ENDTRAP
ENDFUNC file'exists
//
FUNC yes'no$
REPEAT
reply$:=key$
UNTIL reply$>chr$(0) AND reply$ IN "YyNn"
IF reply$ IN "Yy" THEN PRINT "... YES"
IF reply$ IN "Nn" THEN PRINT "... NO"
RETURN reply$
ENDFUNC yes'no$
//
FUNC y'or'n$(state)
IF state THEN
RETURN "YES"
ELSE
RETURN "NO"
ENDIF
ENDFUNC y'or'n$
//
PROC pausekey(tenths)
start'tenths:=computer'tenths
WHILE computer'tenths<start'tenths+tenths DO
reply$:=key$ //reply$ used later
IF esc THEN escape:=true
IF reply$>chr$(0) THEN RETURN
ENDWHILE
ENDPROC pausekey
//
PROC waitkey
clear'keyboard
PRINT AT 0,1: revon$;"Hit <ENTER> to continue";
revoff$, //wrap line
WHILE key$<chr$(1) DO NULL
PRINT AT 0,1: TAB(maxline)
ENDPROC waitkey
//
PROC clear'keyboard
WHILE key$>chr$(0) DO NULL
ENDPROC clear'keyboard
//
FUNC valid(text$) CLOSED
TRAP
dummy:=val(text$)
RETURN true
HANDLER
RETURN false
ENDTRAP
ENDFUNC valid
//
FUNC computer'tenths //c64 users switch // lines
RETURN int(timer*10) // ibm/amiga
//return time div 6 // c64
ENDFUNC computer'tenths
//
===== KEEP PROCS / FUNCS BELOW PURE =====
//
// revision 2.5 sept 17, 1990 - Len Lindsay
FUNC computer'comal
powerdriver:=-1; c64cart:=-2 //Less than 0 is C64/128
unknown'computer:=0 //other than what we can determine
amiga2:=1; cpm2:=2
//future AmigaCOMAL update can go after IBM
ibm2:=3; ibm3:=ibm2 // can't determine ibm 3.0 yet
RANDOMIZE 99 // specific sequence
r'''r:=rnd(1,99) // get the first number

RANDOMIZE // reset pseudo random list
CASE r'''r OF
WHEN 13 // power driver
RETURN powerdriver
WHEN 44 // c64 2.0 cartridge
RETURN c64cart
WHEN 94 // amiga 2.0 plus
RETURN amiga2
WHEN 77 // ibm 2.2 and 3.0
RETURN ibm2 // <<<< all IBM for now
// need a way to distinguish 3.0 from 2.2
WHEN 12 // cp/m 2.10
RETURN cpm2
OTHERWISE // unknown
RETURN unknown'computer
ENDCASE
ENDFUNC computer'comal
//
PROC computer'dims
DIM computer'drive$(0:5) OF 5 //six drives 0,1,2,3,4,5
// end of line delimiter used by system
DIM computer'eol$ OF 2
// reverse field
DIM computer'rvson$ OF 2, computer'rvsoff$ OF 2
// printer name used with SELECT OUTPUT
DIM computer'print$ OF 6
// screen name used with SELECT OUTPUT
DIM computer'screen$ OF 6
ENDPROC computer'dims
//
PROC computer'setup
computer'eol$:=chr$(13)+chr$(10) // CR+LF for IBM,CP/M
computer'print$:="LP:" // default printer
computer'screen$:= "DS:" // screen
FOR x:=0 TO 5 DO computer'drive$(x):=
chr$(x+ord("A"))+":" //ibm & cp/m //wrap line
CASE computer'comal OF
WHEN ibm2,ibm3
computer'rvson$:=chr$(18) // reverse on
computer'rvsoff$:=chr$(19) // reverse off
computer'print$:="PRN:" // default printer
computer'screen$:= "CON:" //screen/keyboard (console)
WHEN amiga2
FOR x:=0 TO 2 DO computer'drive$(x):="DF"+
str$(x)+":" //wrap line
computer'drive$(3):="DHO:"
computer'drive$(4):="DH1:"
computer'drive$(5):="RAM:" // ramdrive
computer'eol$:=chr$(10) // LF (LineFeed)
computer'rvson$:=chr$(16)
computer'rvsoff$:=chr$(16) // toggle rvs
WHEN powerdriver,c64cart // commodore 64/128
computer'eol$:=chr$(13) // CR (CarriageReturn)
computer'rvson$:=chr$(18) // reverse on
computer'rvsoff$:=chr$(146) // reverse off
// only 0 and 1 drives allowed
IF computer'comal=powerdriver THEN
FOR x:=0 TO 5 DO computer'drive$(x):=str$(
(x MOD 2)+":" //wrap line
ELSE //2.0
// unit 8,9,10
FOR x:=0 TO 5 DO computer'drive$(x):=str$(x)+":" //
ENDIF
OTHERWISE // cp/m and unknown
computer'rvson$:=chr$(29)
computer'rvsoff$:=chr$(29) // toggle rvs
ENDCASE
ENDPROC computer'setup
```

CHALLENGE

We hope that many of you will use the \$25 free on-line time on People Link and come to the next few Real Time Programming Challenges (*every first night of each month at 9pm Central!*)! The first correct solution uploaded wins! **The prize is between \$10 and \$50 of free PLink on-line time!** The challenge then continues for the first week of the month, with discussions in the notice area, including helpful tips! To give you an idea of what the Challenge could be, the problems for the last 8 months are listed below. Solutions to these challenges are posted in the COMAL CLUB on People Link (and may be included in the next issue of *COMAL Today*).

April 1990 Challenge

General:

Enter an address and validate the State and ZIP.

Specifics:

Input by user from keyboard

Must accept a NAME

(must have a name, person or company)

Must accept an address

(1 or 2 lines... must be at least one)

Must have a valid USA state 2 letter abbreviation.

MUST have a 5 digit numeric ZIP code.

After the address is entered, the program checks it and replies:

YES if there is a name, at least one address line, valid state, valid ZIP

NO if it does not meet all the criteria for YES

May 1990 Challenge

General:

A computerized shorthand compression. Read in a text file with lines of varying length, but never over 80 characters in one line. Each line ends with CR or CR/LF. Print out the line in compressed form on the screen.

Example:

PROGRAMMING CHALLENGE

might compress to

PRGRMMGCHLLNG

That would be using a rule of dropping all vowels.

Specifics:

- 1) Ask what the maximum length is to be for the output lines.
- 2) Ask for the name of the file to be processed.
- 3) Print to the screen the lines in the file in "compressed form".
 - a) If a line does not exceed the maximum, print it "as is"
 - b) If a line exceeds the maximum, compress it until it is within the max then print it. Object is to have the resulting line be as readable as possible. Possible methods suggested at the meeting were:
 - 1) Delete spaces
 - 2) Delete vowels
 - 3) Delete every other letter
 - 4) Delete z, then q, then ...
 - 5) Replace words with numbers that sound like the word
 - a) TO, TWO, TOO replace with 2
 - b) ATE, EIGHT replace with 8
 - 4) Your program will be tested with a file to be determined later. You can test it yourself with any standard TEXT file.
 - 5) The input text file will have lines delimited by CR or CR/LF. The length of each line will vary, but never will exceed 80.

June 1990 Challenge

General:

Read in a text file and print it one screen at a time. The first screen is printed normally. After that, each subsequent screen is preceded by a "curtain" that falls from the top of the screen to the bottom, "covering" the previous screen of text. Then the next screen of text is revealed from the bottom up to the top, as if the curtain was being raised. Do this till the end of the file.

Specifics:

The input text file will be determined after the entries are all in. It will be an ASCII TEXT file with

CHALLENGE

CR or CR/LF at the end of each line. For testing purposes you may use any ASCII TEXT file. Your program should ask for the name of the text file to use.

The curtain may be displayed as any characters you wish ... be creative.

Don't just erase the text screen and then print the next screen from bottom to the top ... the new text must look as if it is being revealed from behind the rising "curtain".

If your program runs too fast, add some type of delay to give the feeling of a curtain rising and falling.

Options:

Screen sizes may vary, so you may wish to ask how many rows and how many characters per row to use for your screen output.

The length of each line in the text file may vary, and no maximum line length was specified at the challenge. You may wish to allow for lines that are longer than the specified number of characters per output row.

You may have your program automatically do the next screen after a delay, or have it wait for the user to hit RETURN or some other way of indicating to go on to the next screen.

You may wish to allow the user to "abort" out of reading the text file before the end of file is reached.

You may think of other enhancements to make your program entry look and feel the best.

July 1990 Challenge

ENCRYPTION

The idea is to write a program that would take a file and encrypt it so that it is unreadable by a normal text editor or word processor. Then it should also be able to unencrypt the encrypted file back to its original form. Ideally, the same method should work

for ANY file, whether it's a text file, a LISTed COMAL program, or a program or data file, but we will only use a text file to test the entries.

The method of encryption used is totally up to the individual. You can alter each byte somehow, scramble the letters, or ask the user for a password that will have some effect on the encrypted file. That is the method I used in my second entry, but it is all up to you. The only requirements are that the encrypted file should be unreadable when encrypted, and should be able to be restored to its original form.

August 1990 Challenge

General:

Take a text string and print it continuously to the screen in a spiral, starting at the outside of the screen, moving in to the center.

Specifics:

Ask for the screen size (characters per line ... lines per screen). Do not use the last row or the last character on a line.

Ask for the text to print.

Start at the bottom right corner and print counter-clockwise.

Print in a spiral getting smaller till you reach the middle.

Then wait for a key press before the program ends.

Extra Credit:

Allow one space at the end of the text.

Spiral back out from the middle after hitting the middle.

September 1990 Challenge

General:

Take an ASCII text file and reformat it so that each line has a max of 79 characters and will print on 40 column screens too (without splitting words).

Continued...

CHALLENGE

...Continued

Specifics:

Input file is an ASCII text file, such as one produced by your word processor. Our problem was to remain flexible, and accept a file with end of line delimiters at the end of each line in a paragraph, or just at the end of the paragraph. Beginners should write a solution just for their word processor output type (one or the other).

The program should work with files from any computer, so the end of line delimiters could be:

CR
LF
CR/LF

NOTE: CR=Carriage Return
LF=Line Feed

Beginners should write a solution just for the type of delimiter used by their word processor output file (CR/LF is usual for IBM, LF is usual for Amiga).

The output file should be suitable for on-line viewing on PLINK. This means each line must be 79 characters or less and end with a CR.

If a line is 40 characters or less, output it as is. Otherwise, add spaces evenly between words so that there is a space at column 40 and NOT a space at column 39. Then finish off the line to a max of 79 characters.

Extra Credit:

Full justification of the whole line.

October 1990 Challenge

Last month was a hard challenge ... to make up for it, this month the challenge is easy.

Just generate 6 random numbers between 1 and 44 inclusive ... then print them on the screen in ascending sorted order. Yes ... it is a lottery number generator.

Extra Credit:

Make sure no number is duplicated.

November 1990 Challenge

General:

Write a program that asks for a phone number and helps you find a cute "word" that could be dialed to get that phone number.

Specifics:

- 1> Program verifies phone number input at keyboard as proper
- 1a> First digit cannot be a 0 or a 1
- 2> Phone number must be 7 digits
- 3> After entering the phone number the program starts cycling
- 3a> Cycles through all valid letters for the digit, plus allows the number
- 4> Program lets user do one or more of the following:
 - 4a> Program may allow user to FREEZE and UNFREEZE digits
 - 4b> Program may let user assign a letter/number to a digit (when assigned, it is automatically FROZEN)
 - 4c> THUMBWHEELS type action on digits is acceptable
- 5> Program lets user PRINT current WORD *OR* STORE it to a disk file
- 5a> Can be stored at any time
- 6> Program lets you speed up or slow down the cycling
- 7> Cycling is changing a digits current letter/number
- 7a> If it is a 2 ... cycling would make it 2 then A then B then C then 2 ...
- 7b> Cycling can be forward or backward as long as all possible combinations
- 8> Program must be able to show ALL POSSIBLE combinations

Phone Digit Facts:

1 ... 1	6 ... 6 M N O
2 ... 2 A B C	7 ... 7 P R S (no Q)
3 ... 3 D E F	8 ... 8 T U V
4 ... 4 G H I	9 ... 9 W X Y (no Z)
5 ... 5 J K L	

People Link Overview

by David Warman

American PeopleLink, more commonly known as PLink, is a national computer network with international access that has a lot to offer. In addition to the usual EMail, "party lines" for general conversation, travel services, etc., PLink has many Clubs dedicated to special interest groups. Among them are the COMAL Club, which I think has the potential to become the best on-line COMAL support club on any network. Amiga users also have access to the Amiga Zone, the biggest and most active on-line Amiga club anywhere.

PLink has reasonable rates when accessed via Redi-Access; \$4.95 an hour at 300 baud, \$5.95 at 1200 baud, and \$9.95 at 2400 baud during non-primetime hours (6:01 PM to 7 AM your local time, and all day on weekends and some holidays). Telenet's rates are slightly higher, but by using a PCPursuit account you can get rates as low as \$3.50 an hour at any baud rate. Become a Frequent PLinker (\$10 per month) and take 25% more off that! [prices are set by PLink and subject to change.]

You can access PLink with virtually any terminal program on any computer, and can upload or download files with XModem or the faster WXModem protocol.

To sign up with PLink call **1-800-826-8855** with your modem or **1-800-524-0100** voice. Make sure you say you heard about it from COMAL to get \$25 free on-line time. You can have your account set up to charge your credit card or you may write checks for the amount you want added to your PLink account. Once you get your account established, you will need to find your local Redi-Access or Telenet number. A list of these numbers begins on page 57.

If you call via Redi-Access, after you dial your local number and are connected, you get a blank screen. Wait 1 second, then type a period 4 times (...) followed by the <enter> key, with a one second pause between each (.). If it doesn't work, try again, Redi-Access is sometimes picky about the timing. At the ==> prompt, type PLINK <enter> and you'll be almost instantly connected to PLink.

If you are calling via TeleNet (now SprintNet), it is even easier. I have never used TeleNet, but from what I've been told, for 2400 baud access you type: @D <enter> or for 300/1200 baud access you type: <enter> D <enter>. Wait for the **TERMINAL=** prompt, then type: D1 <enter>. Wait until you get the "@" prompt, then type C PLINK <enter> and you will be connected to PLink. If you use a PCPursuit account, the procedure will be somewhat different, as you will have to enter your account number and password, but PCPursuit should supply you with that information.

Once you are connected to PLink, you will be asked to enter your PLink ID and password, which was supplied by PLink when you signed up. Then PLink will display some brief signon announcements, after which you can do whatever you want. The ID PLink assigns to you is a "generic" ID, such as OES917. You can get it customized to any name you wish up to 10 characters that is not already taken (such as CAPTAIN C or DAVID*W). To do this just type: /GO 411

The /GO command is the fastest and easiest way to get around PLink. The slash before PLink commands is usually optional, but I'll use them here to make the commands stand out. When you first join PLink, you may want to use /MENU mode (*this is the default mode, or later type: /MENU at a main prompt*) which lists most possible commands that you can enter at a given time and allows you to select from them using numbers. Once you become more experienced, you can switch to the more convenient /ADVanced mode, which removes the numbered menu. Type /ADV to switch. You still can get a menu of choices at any time by typing: ? Note that most commands may be typed in full /MENU or abbreviated to the first three letters /MEN.

PLink's main features are the Clubs, all of which operate similarly. To get to the COMAL Club, just type /GO COMAL from virtually anywhere on PLink. The first time you enter the Club you will be asked if you want to join. You can read messages in the COMAL Club without joining, but you will need to join to enter the conference rooms and to get into certain sections of the Club. To join (*which costs absolutely nothing*) just reply Y to the prompt,

Continued...

People Link Overview

...Continued

or type /JOIN at any later time while in the COMAL Club.

The COMAL Club is staffed by:

CAPTAIN C	Len Lindsay, chairman
DAVID*W	David Warman, asst.
M*PILCHUK	Mitch Pilchuk, asst.
COMALTODAY	Joel Rea, asst.
G*TURNEY	Geoffrey Turney, asst.

The COMAL Club is divided into 15 sections; each section has a Notice board and most have a Data Library (upload/download area). There are individual sections for News, Tutorials, Commodore 64, Amiga, IBM, and other versions of COMAL. We also feature a Workshop where anyone may upload a program (**upload time is free!**) to get help on a programming problem. If you upload it in LISTed format (*with Carriage Returns at the end of each line*) it can be read on-line or downloaded and ENTERed into any version of COMAL.

The COMAL Club is the host of a monthly **Real Time Programming Challenge**, an exciting contest that is open to ANY computer, ANY language. On the first night of each month at 9 PM Central we meet briefly in the COMAL Conference room on line 1 to decide on the goal of the Challenge (/CON;LIN 1). Anyone present may make suggestions or modify suggestions made by others. Then we break up to work on the solution (which should be something that can be accomplished in about an hour). The **first** correct upload is declared the winner. One month the first two entries uploaded were a mere one minute apart! These Challenges are a good way to have fun and improve your programming skills at the same time. Afterwards, you can compare your entry to the others, and get helpful advice for improvements. So far every challenge has been won by one of the COMAL Club staff members. You can try to solve the next Challenge faster than the COMAL Club staff ... or take up to a whole week to work out a nice solution. After one week all entries are posted for all to see and test. **The first correct uploaded solution wins a prize of free PLink on-line time (usually between \$10 and \$50).**

Below is a brief list of some of PLink's commands, to help new users get around.

/MAIL - Send EMail while in the COMAL Club. Same options as with /GO MAIL

/GO MAIL - This takes you to The EMail section. Once there, type /SEND to start a message to someone. Enter their ID name and a SUBJECT. Then enter the message (you can prepare the message off-line and ASCII SEND it over to PLINK at this point). To end your message type: /END. Then type: /SEND again to actually send the EMail (it is common for PLink to require you to repeat your original command when through with the task). Or type /HELP to get more information and a list of mail options. These are some of the option that you can add before you type the final /SEND:

- /AR - You will be notified when the person reads your EMail
- /EXP - Express Mail (*extra cost*)
- /CC NAME - Send carbon copy to NAME. This CC appears at bottom of the EMail
- /BCC NAME - Send carbon copy to NAME. Nothing shows at bottom of EMail (it is a "blind" carbon copy). Useful to send a copy of EMail to yourself.

/GO 411 - Information. One option here is to change your screen ID from the WCS123 type of ID PLink assigns you when you first sign up. Go here to change your ID to a more personal ID. Give yourself any name up to 10 characters long that is not already being used by someone else.

CLUB COMMANDS

- /TOP - Jump to the top of the section.
- /LIB - Enter the Library part of a Club.
- /NOT - Enter the Notices part of a Club.
- /CON - Enter the Conference part of a Club.
- /NOMore and /NOPause get rid of the sometimes annoying Press <return> for more prompts

People Link Overview

and eliminate pauses between messages. You can use **CTRL-s** and **CTRL-q** to pause and resume at your own speed.

/SEC - Choose a section from 1-15. You can also choose section **ALL** for reading and scanning (*the default*), but must choose a single section for posting a message or file.

/OPT - Choose your default options for a Club, so you don't have to set them every time you sign in. These options include the default section you are in when you enter the Club, and the method used to read messages: forward, backwards, dialogue order, etc. You can also type **/GO DEFAULT** to permanently change other settings.

/READ ALL - Read all messages in the current section (which can be section **ALL**). If you are in section **ALL** you will be able to read all messages, but if you are in section 1, for example, the **READ ALL** command will only display messages in section 1.

/READ NEW - Read messages in the current section that were posted after the last message you have read in the section.

/SCAN ALL & /SCAN NEW - Scan all or new messages, which just shows the headers, without the text.

/QSC ALL & /QSC NEW - Briefe Quick SScan.

These reading and scanning commands work the same way in the Libraries as in the Notices section, except that in the Libraries you will be reading file descriptions instead of messages. You can precede each of these commands with an **R** and the messages will be shown in reverse date order (**/RQSC /RSCAN /RREAD**).

/POST - Start posting a message or uploading a file (depending on whether you are in the Notice boards or the Libraries). Next, type in your message (or file description) and type **/POST** again to indicate that the message is complete. (For more detailed information on uploading and downloading

files, there is a file in SEC 2 of the COMAL Library called **UPLOADS.TXT** which you can either read on-line or download once you have joined the Club.)

There are many other options available, such as reading the messages in **/FORward** or **/REVerse** order, and by **/DIAlog** (thread order) or **/NODialog** (time order). You can also read messages **/TO** or **/BY** a particular person, or those posted **/SINce** a specific day. And you can combine all of these options to narrow your reading list to a very exact group of messages.

One thing I really like about PLink is that you can type a message to someone off-line using your favorite word processor, then use your terminal program's ASCII SEND mode (if it has one) to send the message very quickly, thus saving on-line time and allowing you to edit your letter before sending it. PLink's on-line editor is OK, but a word processor is much better. To send a pre-written message, you just type **/POST** to start a message in the usual way, then instead of typing (or BEFORE or AFTER or IN BETWEEN typing something) go into ASCII SEND mode on your terminal program and your message will be quickly zipped along to PLink, just as if you had typed it in. Finally, type **/POST** again to post your message. If you forget the second POST the message will be lost.

Likewise, you may ASCII SEND EMail and file descriptions in the same way. One thing to remember when using this method is that PLink expects every line you send to end with a carriage return (or line feed, or whatever your computer uses) before the 80'th character. Some word processors can end each line with a CR, but even if yours doesn't you can still ASCII-send your message; it may have words broken on the end of the line, but it should still be readable, at least by terminal programs that have a word-wrap feature. Of course, your word processor must be able to output straight ASCII (or PetASCII) without any imbedded formatting commands.

PLink Voice Info: 1-800-524-0100

PLink Modem Info: 1-800-826-8855

"Where did you hear about us?" →COMAL

People Link Tips

by Mitch Pilchuk

☞ Usually anytime the /NEXT command option is called for, such as for the next message, you can press <Enter> to accomplish the same thing.

☞ You actually can get away with only typing in the first three character of most commands. Example: /DLOAD would be /DLO.

☞ You can use UPPER or lower case letters when typing commands.

☞ The / prior to any command can be left off in the most cases. It must be used in the conference area. When writing a Notice, EMail, or file description you need to type the slash before END, EDIT, SEND, DIS, ULO, etc. Generally, you need to use the slash whenever you want to type a command, but you are not at a PLINK prompt.

☞ You can change your ID ("handle") on PLINK once for free. A personalized ID will make it easier for others to remember you. Just type: /GO 411. COMAL Club staff includes:

CAPTAIN C	- Len Lindsay, chair
DAVID*W	- David Warman, assistant
M*PILCHUK	- Mitch Pilchuk, assistant
COMALTODAY	- Joel Rea, assistant
G*TURNERY	- Geoffrey Turney, assistant

☞ The advanced mode is the best mode to use. You can download a text file (Amiga Zone file 20572) that is a Peoplelink manual. It has all of the commands in it. Get this file, print it out and keep it nearby for reference. Menu mode is cumbersome and doesn't really help you learn how to navigate this network. It's good in a pinch, but that's all.

☞ You can put more than one command on a line by separating each command with a semi-colon or space. Example, the following:

/DLOAD
/FILE
/XMODEM

could be typed:

/DLO;FIL;XMO <return>

☞ Use the Nopause command (/NOP) to turn off the pause between messages when you want to

capture the messages without reading. Pause (/PAU) turns it back on.

☞ Likewise, use the /NOM command (nomore) to get rid of the "Press RETURN for more prompt".

☞ There is no auto wrap at the end of each message line, so when entering a message you must press the return key to end the sentence and start a new line. You should press RETURN or ENTER on or before the 79th character (*Baud Bandit term program has a feature that gives you word wrap. Look in the term programs manual.*)

☞ You can send EMail from inside the COMAL Club. Just type: /MAIL. You can send an EMail to more than one person... just put multiple IDs at the SEND TO: prompt (CAPTAIN C,DAVID*W). You also can carbon copy others ... meaning a copy of the EMail will be sent to them to. To do that issue the command: /CC NAME ... example:

/CC CAPTAIN C,DAVID*W

And did you ever wonder if the person even gets your EMail? Just issue the command /AR before your final /SEND and you will receive a notice when they read the EMail.

☞ Wonder if someone is signed onto PLINK? Just send them a message. If they are signed on you will get a =MESSAGE TRANSMITTED=, otherwise you will get a message that the person is not signed on! Just do this: /MSG DAVID*W <ENTER> ... then type a message, such as "HI".

☞ You can see who else is using the CLUB you are currently in. Just type: /LIST.

☞ There is a special USERS DIRECTORY ... listing user IDs and their interests. Put yourself in it (make sure you include COMAL as an interest). Type: /GO UD and then follow the prompts.

PLink Meeting Schedule (9pm Central)

First night each month - Fun Challenge

First Wednesday each month - meeting

Third Thursday each month - meeting

People Link Walk Thru

by Mitch Pilchuk

This is an example of a sample session on People Link. Things you type are underlined. Comments are inside {}.

CONNECTED TO PEOPLE/LINK!

USER ID: CAPTAIN C
PASSWORD: *****

Welcome to PEOPLE/LINK, CAPTAIN C

Today is Saturday, 20 October 90, 11:41:05
Last visit Saturday, 20 October 90, 08:45:01

PLINK NEWS FLASHES APPEAR HERE!

{this is in MENU mode, later we switch to ADVanced mode}

PEOPLE/LINK Main Menu

- | | |
|-------------------------|----------|
| 1 PartyLine | /PARTY |
| 2 Clubs & Forums | /CLUBS |
| 3 Online Shops | /SHOPS |
| 4 Mail | /MAIL |
| 5 Travel | /TRAV |
| 6 User Directory | /UD |
| 7 Bulletin Boards | /BB |
| 8 Information | /GO 411 |
| 9 News and Publications | /GO NEWS |

Enter number, command or /HELP
MAIN MENU> /go comal {go to the comal club}

Welcome to People/Link Clubs!

COMAL Club welcomes you!!!

You were last in the club on
20 October 1990 at 08:59.

WELCOME TO THE COMAL CLUB

COMAL is a programming language that is designed for beginners... but powerful enough for the professional programmer. It is standardized and available for: IBM PC & PS/2, C64, C128, Amiga & CP/M. COMAL INFO is in the library. Find the files with this command: /QSC KEY INFO

The first night of each month we have a REAL TIME PROGRAMMING *FUN* CHALLENGE. It starts at 9pm Central in CONF LIN 1. Any computer. Any language. Everyone's invited. See SEC 12 for details. Past Challenge entries are in the library.

MEETINGS (/CONF;LIN 1) (Central time)

Every first WEDS of the month at 9 p.m.
Every third THUR of the month at 9 p.m.

***** FREE UPLOADS TODAY *****

Your SETTING (Default) is to read ALL sections.

CLUB MENU

- | | |
|-----------------------------|-------|
| 1 Message And Notice Boards | /NOT |
| 2 Library Of Transfer Files | /LIB |
| 3 Club Conference Area | /CONF |

- | | |
|---------------------------|----------|
| 4 CLUB SELECTION Menu | /GO CLUB |
| 5 Club Information | /INFO |
| 6 Exit Clubs To MAIN Menu | /QUIT |

Please enter number or command:
CLUBS> /list {list everyone in the club now}
(note that ALL commands were not shown in the MENU)

Currently using the COMAL Club
CAPTAIN C M*PILCHUK

CLUB MENU

- | | |
|-----------------------------|----------|
| 1 Message And Notice Boards | /NOT |
| 2 Library Of Transfer Files | /LIB |
| 3 Club Conference Area | /CONF |
| 4 CLUB SELECTION Menu | /GO CLUB |
| 5 Club Information | /INFO |
| 6 Exit Clubs To MAIN Menu | /QUIT |

Please enter number or command:
CLUBS> /not {move to the NOTICE area}

You are now in the COMAL Club notice area!

CLUB NOTICE BOARD MENU

- | | |
|-----------------------------|---------|
| 1 Select Club Section | /SECT |
| 2 QuickScan Notice Headings | /QSCAN |
| 3 Read Notices | /READ |
| 4 Post A Notice | /POST |
| 5 Change Notice You Posted | /CHANGE |
| 6 Delete Notice You Posted | /DELETE |
| 7 Return To CLUB Menu | |
| 8 Exit Clubs To MAIN Menu | /QUIT |
| 9 Info On These Options | /HELP |

Please enter number or command:
CLUB NOTICES> /sec
(look at what sections are available)

COMAL Club's sections are:

1. News & Announcements (Read Only)
 2. Resources (Read Only)
 3. Tutorials
 4. General
 5. Amiga
 6. Commodore 64 / 128
 7. IBM PC, PS/2, MS-DOS
 8. CP/M, Apple, Mac, Other
 9. New User Questions (Potential Members)
 11. WorkShop
 12. CHALLENGE
 13. Fish Disks (Amiga Sideline)
- Please enter a section number or ALL
> all {the same section we started in...all sections}

You have just entered the
COMAL Club , Section ALL, in
read/scan-only mode.

CLUB NOTICE BOARD MENU

- | | |
|-----------------------------|---------|
| 1 Select Club Section | /SECT |
| 2 QuickScan Notice Headings | /QSCAN |
| 3 Read Notices | /READ |
| 4 Post A Notice | /POST |
| 5 Change Notice You Posted | /CHANGE |
| 6 Delete Notice You Posted | /DELETE |
| 7 Return To CLUB Menu | |
| 8 Exit Clubs To MAIN Menu | /QUIT |
| 9 Info On These Options | /HELP |

Continued...

People Link Walk Thru

...Continued

Please enter number or command:
CLUB NOTICES> /read {read messages}

YOU MAY READ:

- | | |
|------------------------------|--------|
| 1 All Notices | /ALL |
| 2 New Notices Only | /NEW |
| 3 Old Notices Only | /OLD |
| 4 Notice With Certain Number | /NUM |
| 5 Notices By Certain User ID | /BY |
| 6 Notices To Certain User ID | /TO |
| 7 Notices With Certain Title | /TITLE |
| 8 Notices Since MM/DD/YY | /SINCE |
| 9 Return To NOTICE Menu | /TOP |
| 10 Info On Reading Notices | /HELP |

Please enter number or command:
READ NOTICES> /sin {lets read since a date}
Enter date as MM/DD/YY
-> 10/01/90
(earlier we could have typed: /READ SIN 10/01/90)

Club : COMAL Sec: 12
Date : 10/02/90 2:10 Num: 1,602
Theme: OCT 1990 CHALLENGE
To : ALL By : CAPTAIN C
Title: OCT 1990 CHALLENGE

Last month was a hard challenge ... to make up for it, this month the challenge is easy. Just generate 6 random numbers between 1 and 44 inclusive ... then print them on the screen in ascending sorted order. Yes ... it is a lottery number generator. Extra credit ... make sure no number is duplicated.

Upload your solution to the library. You have one week to do so.

As always ... solutions can be in any language ... for any computer. Prize to the first correct uploaded solution this month is \$20 of free PLINK on-line time.

AFTER READING A NOTICE, YOU MAY:

- | | |
|--------------------------------|---------|
| 1 Read Notice Again | /AGAIN |
| 2 Reply To The Notice | /REPLY |
| 3 Read Reply To This Notice | /RREPLY |
| 4 Read This Dialog From Top | /RTOP |
| 5 Read Next Consecutive Notice | /SKIP |
| 6 Read Next Notice | /NEXT |
| 7 Return To NOTICE Menu | /TOP |
| 8 Info On Reading Notices | /HELP |

Please enter number or command:
NOTICE RESPONSE> /top {back to top of area}

CLUB NOTICE BOARD MENU

- | | |
|-----------------------------|---------|
| 1 Select Club Section | /SECT |
| 2 QuickScan Notice Headings | /QSCAN |
| 3 Read Notices | /READ |
| 4 Post A Notice | /POST |
| 5 Change Notice You Posted | /CHANGE |
| 6 Delete Notice You Posted | /DELETE |
| 7 Return To CLUB Menu | /QUIT |
| 8 Exit Clubs To MAIN Menu | /QUIT |
| 9 Info On These Options | /HELP |

Please enter number or command:
CLUB LIBRARY> /conf {switch to the conference area}

Welcome to COMAL conference!

There are no active public conference lines at this time.

CLUB CONFERENCE MENU

- | | |
|------------------------------|---------|
| 1 Show Conf Lines In Use | /SUM |
| 2 List Members In Conference | /LIST |
| 3 Join Conference | /LINE |
| 4 Request Private Chat | /CHAT |
| 5 Join Private Conference | /CODE |
| 6 Change Your Handle (Name) | /NAME |
| 7 Add'l Conference Features | |
| 8 Exit To Club Notice Boards | /QUIT |
| 9 Exit To MAIN Menu | /GO TOP |
| 10 Info On These Options | /HELP |

Please enter number or command:
CLUB CONF> /lin 1 {enter Line 1}

You entered Line 1. {partial conference example below}

(CBM*HARV) If Mark wins, we'll never hear the end of it.
(INFO MAG) In INFO: Editor Wins Contest! Film At 11. ;)
(CAPTAIN C) OK ... is it time to say READ SET GO?
(DAVID*W) Ready.

(CBM*HARV) Copperman to award prize in Iowa City Parade:
CBM Plus/4

(INFO MAG) I already have one!
(CAPTAIN C) Winner will get a prize ... all entrants get a consolation prize if they ask for one.
(INFO MAG) Autographed copy of very Amish-looking Wisconsin gentleman! :)

/QUIT {gets you out of the conference}

Exiting COMAL conference - see ya later!

Welcome to People/Link Clubs!

COMAL Club welcomes you back!!!

CLUB MENU

- | | |
|-----------------------------|----------|
| 1 Message And Notice Boards | /NOT |
| 2 Library Of Transfer Files | /LIB |
| 3 Club Conference Area | /CONF |
| 4 CLUB SELECTION Menu | /GO CLUB |
| 5 Club Information | /INFO |
| 6 Exit Clubs To MAIN Menu | /QUIT |

Please enter number or command:
CLUBS> /adv {switch to ADVANCED mode - no menus}

Please enter Command or /HELP:

CLUBS> /read all

{read all messages (we are in NOTICE area by default)}

Club : COMAL Sec: 5
Date : 3/20/90 17:45 Num: 328
Theme: FUNCTION KEYS UPDATE
To : M*PILCHUK (r) By : DAVID*W
Title: R#326 LOOKS RIGHT

Yep, that looks right, if I remember correctly. (I'm on the C64 right now, so I can't check.) I know the values \$80 to \$89 are correct, as I've written a program that looks for those keystrokes.

Continued...

People Link Walk Thru

...Continued

I also found out that if you use Setkey to define F1 to equal LIST, for example, pressing F1 will return the values \$4c, \$49, \$53, and \$54 in succession. (Again, those numbers are off the top of my head. What I mean is that the ASCII values of L,I,S, &T are ALL returned.) However, if you use the normal keymap, COMAL will read the value \$80 from F1.

-David-

/NEXT /SKIP /RREPLY /RTOP /REPLY /AGAIN
-> /top {quit reading messages, jump to top of area}

Please enter Command or /HELP:
CLUBS> /lib {go to the Library files area}

You are now in the COMAL Club library!

Please enter Command or /HELP:
CLUBS LIB> /qsc {quick scan of files available}
/ALL /NEW /SIN /BY /NUM /TITL /KEY /NAME
LIB> /sin {choose SINCE a date}

Enter date as MM/DD/YY

-> 5/15/90 {we could have typed: /QSC SIN 5/15/90}
1) MITCH.LST By: CAPTAIN C 5/16/90 Downloads: 1
MITCH PILCHUK LATE ENTRY Sec: 12 Num: 256 3.8KB
2) CONF-MAY.TXT By: DAVID*W 5/18/90 Downloads: 5
MAY 1990 COMAL CONF TRANSCR Sec: 2 Num: 258 11.8KB
3) BUFFER/PWR.SDA By: DAVID*W 5/19/90 Downloads: 1
POWER DRIVER BUFFER ROUTINE Sec: 6 Num: 259 14.3KB
4) CHALLENG.MAY By: DAVID*W 5/23/90 Downloads: 9
CHALLENGE TRANSCRIPT TEXT Sec: 12 Num: 266 38.1KB

Enter number(s) separated by commas or
press <return> for more choices:

LIB> 4 {take a look at the description of item 4}

COMAL Sec: 12
Name : CHALLENG.MAY Num: 266
By: DAVID*W Date: 5/23/90 22:09
Bytes: 38,144 Downloads: 9
Binary!
Title: CHALLENGE TRANSCRIPT TEXT FILE
Keywords: TRANSCRIPT CHALLENGE MEETING CONFERENCE TEXT

This is the transcript of the May 1990 Challenge
meeting. Those who still want to enter the Challenge for
fun can read this file to find out what problem is being
solved. To read it on-line, just type /DLO;FIL;ASC at
the next prompt. You do not have to type the commands
in caps.

Good luck in the Challenge!

/NEXT /DLOAD /AGAIN /TOP
LIB> /top {skip downloading, jump back to top of area}
Please enter Command or /HELP:
CLUBS LIB> /menu {turn the menus back on}
MENUS ARE ON!

CLUB DATA LIBRARY MENU

- | | |
|---------------------------------|-------|
| 1 - Enter a Library section | /SEC |
| 2 - QuickScan files & download | /QSC |
| 3 - Read description & download | /REA |
| 4 - Add a file to the Library | /POST |
| 5 - Delete a file you added | /DEL |
| 6 - List the Library files | /XFER |
| 7 - Return to the Notice Menu | /NOT |
| 8 - Return to Club Main Menu | /QUIT |
| 9 - Exit Clubs | /HELP |
| 10 - More information | |

Please enter number or command:

CLUB LIBRARY> /go mail
(go the EMAIL section of PLINK, leave COMAL Club)

= MAIL MENU =

- | | |
|--------------------------------|--------|
| 1 - Read Mail | /READ |
| 2 - Scan Mail Headings | /SCAN |
| 3 - QuickScan Mail Headings | /QSCAN |
| 4 - Delete Unwanted Mail | /DEL |
| 5 - Write And Send Letter | /SEND |
| 6 - Retrieve Old Mail & Resend | /RET |
| 7 - Use Advanced Mode | /ADV |
| 8 - Exit Mail To MAIN Menu | /QUIT |
| 9 - Information About Mail | /HELP |

Please enter number or command:

MAIL> /read {usually you just type: /READ NEW}

YOU MAY READ:

- | | |
|------------------------------|-------|
| 1 - All Mail | /ALL |
| 2 - Unread (New) Mail Only | /NEW |
| 3 - Specific Letter Number | /NUM |
| 4 - From Specific Sender | /FROM |
| 5 - Mail on Specific Subject | /SUBJ |
| 6 - Mail of a Specific Date | /DATE |
| 7 - Return to MAIL Menu | /TOP |
| 8 - Info about Reading Mail | /HELP |

Please enter number or command:

READ MAIL> /new

Date 10/20/90 11:21
From DAVID*W
Subj STATS
Num 966

Folder Name: INBASKET

{Sorry, this is private mail ... a letter was here}

AFTER READING LETTER, YOU MAY:

- | | |
|------------------------------|--------|
| 1 - Read Letter Again | /AGAIN |
| 2 - Answer Letter | /ANS |
| 3 - Download Attached File | /DLOAD |
| 4 - Delete This Letter | /DEL |
| 5 - Forward Letter | /FOR |
| 6 - Read Next Letter | /NEXT |
| 7 - Return to MAIL Menu | /TOP |
| 8 - Info about These Options | /HELP |

Please enter number or command:

MAIL RESPONSE> /off {sign off People Link}

Signed off at 11:56:05

Connected 000 hours 15 minutes.

THANK YOU FOR VISITING PEOPLE/LINK!

Remember ... sign up to People Link and mention
COMAL to get a free \$25 of on-line time! See
their ad in this issue for info on this special offer.
One of the first things you should do on-line is
/GO 411 and create a personalized ID for
yourself. Then /GO COMAL and post a message!

**PeopleLink Access Numbers
(in Area Code order)**

r = Redi-Access
t = TeleNet 300/1200
* = TeleNet 300/1200/2400
+ = TeleNet 2400
= TeleNet 300

(201)

417-9445 r NJ Edison
780-5030 t NJ Freehold
780-9122 + NJ Freehold
488-2063 + NJ Hackensack
488-6567 t NJ Hackensack
644-4745 + NJ Morristown
455-0275 t NJ Morristown
366-1560 r NJ Morristown
745-2900 t NJ New Brunswick
745-7010 + NJ New Brunswick
623-0469 t NJ Newark
623-7122 + NJ Newark
773-3674 + NJ Passaic
778-5600 t NJ Passaic
684-7560 t NJ Paterson
742-4415 + NJ Paterson
815-1885 t NJ Rahway
571-0003 t NJ Redbank
227-5277 t NJ Roseland
227-6722 + NJ Roseland
525-9507 t NJ Sayreville
774-1642 r NJ Shrewsbury
273-9619 * NJ Summit
669-8011 r NJ W Orange

(202)

429-7800 t DC Washington
429-0956 + DC Washington
429-7896 t DC Washington

(203)

579-1122 r CT Bridgeport
335-5055 t CT Bridgeport
367-9130 + CT Bridgeport
792-5354 + CT Danbury
794-9075 t CT Danbury
748-2648 r CT Danbury
629-2070 r CT Greenwich
247-9479 t CT Hartford
724-9396 + CT Hartford
659-9636 r CT Hartford
344-8217 t CT Middleton
225-7027 t CT New Britain
624-5954 t CT New Haven
234-2000 r CT New Haven
773-3569 + CT New Haven
437-0909 + CT New London
442-7245 r CT New London
447-8455 t CT New London
866-7404 t CT Norwalk
853-0179 r CT Norwalk
359-9404 + CT Stamford
348-0787 t CT Stamford
753-4512 t CT Waterbury
756-0342 + CT Waterbury
456-4143 r CT Willimantic

(205)

236-9711 t AL Anniston
821-5351 r AL Auburn
328-2310 t AL Birmingham
251-1885 + AL Birmingham
942-9301 r AL Birmingham
355-0206 t AL Decatur
793-5034 t AL Dothan
767-7960 t AL Florence
767-0497 + AL Florence
837-4050 r AL Huntsville
539-1631 + AL Huntsville
539-2281 t AL Huntsville

660-0622 r AL Mobile
438-6881 + AL Mobile
432-1680 t AL Mobile
269-0085 r AL Montgomery
832-4314 + AL Montgomery
269-0090 t AL Montgomery
752-6968 r AL Tuscaloosa
758-5799 + AL Tuscaloosa
752-1472 t AL Tuscaloosa

(206)

939-9982 t WA Auburn
733-2720 t WA Bellingham
733-3421 r WA Bellingham
733-2873 + WA Bellingham
774-7466 + WA Everett/Lynwd
775-9929 t WA Everett/Lynwd

(215)

577-3992 + WA Longview
577-5835 t WA Longview
774-7466 * WA Lynwood
357-7499 r WA Olympia
754-0460 t WA Olympia
786-5066 + WA Olympia
625-9612 t WA Seattle
623-9951 + WA Seattle
455-9260 r WA Seattle
627-1791 t WA Tacoma
572-4125 r WA Tacoma
693-6914 t WA Vancouver

(216)

253-3945 r OH Akron
453-7552 r OH Canton
452-6642 + OH Canton
452-0903 t OH Canton
226-1776 r OH Cleveland
575-1658 t OH Cleveland
771-6480 + OH Cleveland
941-0844 r ME Bangor
989-3081 t ME Brewer
783-1353 r ME Lewiston
784-0105 t ME Lewiston
773-9846 r ME Portland
761-4000 t ME Portland

(208)

336-8540 r ID Boise
343-1272 + ID Boise
343-0611 t ID Boise
529-0406 * ID Idaho Falls
746-6946 r ID Lewiston
743-5885 + ID Lewiston
743-0099 t ID Lewiston
232-1764 * ID Pocatello

(209)

441-1861 + CA Fresno
233-0961 t CA Fresno
233-1291 r CA Fresno
383-2557 t CA Merced
576-2852 t CA Modesto
576-2451 r CA Modesto
944-5947 r CA Stockton
957-7610 t CA Stockton
957-7657 + CA Stockton
627-1201 t CA Visalia

(212)

956-4244 r NY New York
956-4271 r NY New York
741-8100 t NY New York
645-0560 + NY New York

(213)

516-1007 t CA Compton
337-1187 r CA Inglewood
428-9140 r CA Long Beach
624-2251 t CA Los Angeles
480-0734 r CA Los Angeles
622-1138 + CA Los Angeles
306-2984 t CA Marina Del Ray
306-4922 + CA Marina Del Ray
404-2237 t CA Norwalk
548-6141 t CA San Pedro
514-1590 + CA San Pedro
458-6598 r CA Santa Monica

(214)

677-1712 t TX Athens

788-4651 r TX Dallas
748-6371 t TX Dallas
745-1359 + TX Dallas
753-1929 r TX Longview
236-4205 t TX Longview
893-4995 t TX Sherman
592-6544 r TX Tyler
597-8925 t TX Tyler

(215)

770-1405 + PA Allentown
435-3330 t PA Allentown
974-9400 r PA Bethlehem
337-4300 t PA King of Prusia
337-2850 + PA King of Prusia
692-2796 r PA King of Prusia
946-3469 * PA Levittown

(304)

854-8121 r PA Philadelphia
574-9462 t PA Philadelphia
574-0990 + PA Philadelphia
375-6945 + PA Reading
374-2271 r PA Reading
376-8750 t PA Reading
343-6010 * PA Warrington

(216)

253-3945 r OH Akron
453-7552 r OH Canton
452-6642 + OH Canton
452-0903 t OH Canton
226-1776 r OH Cleveland
575-1658 t OH Cleveland
771-6480 + OH Cleveland
322-8712 t OH Elyria
322-8645 + OH Elyria
678-5115 t OH Kent
678-5043 + OH Kent
960-1771 t OH Lorain
393-0111 r OH Warren
394-0041 t OH Warren
264-8920 t OH Wooster

(217)

743-6843 + OH Youngstown
743-1296 t OH Youngstown
746-7200 r OH Youngstown

(218)

398-1712 r IL Champaign
422-4012 r IL Decatur
429-0235 t IL Decatur
429-6054 + IL Decatur
753-1373 t IL Springfield
522-1549 r IL Springfield
753-1391 + IL Springfield
384-6428 t IL Urbana
328-0317 + IL Urbana

(219)

723-8005 r MN Duluth
722-5032 + MN Duluth
722-1719 t MN Duluth

(210)

422-3431 + IN Fort Wayne
426-2268 t IN Fort Wayne
484-3963 r IN Fort Wayne
736-2708 r IN Gary
882-8800 t IN Gary
882-1835 + IN Gary

(211)

234-9938 r IN South Bend
233-7104 t IN South Bend
233-4031 + IN South Bend

(301)

224-0795 + MD Annapolis
626-1451 r MD Annapolis
224-8550 t MD Annapolis
752-5555 + MD Baltimore
727-6060 t MD Baltimore
281-9222 r MD Baltimore
724-1775 r MD Cumberland
293-9596 t MD Frederick
369-3482 r DC Washington

(302)

678-8328 t DE Dover
737-4340 + DE Newark
454-7710 t DE Newark
454-7834 r DE Wilmington

(303)

741-5612 r CO Denver
337-6060 t CO Denver
696-0159 + CO Denver
493-9131 t CO Fort Collins
484-3978 r CO Fort Collins
493-4014 + CO Fort Collins
337-4300 t PA King of Prusia
241-3004 t CO Grand Junction
337-2850 + PA King of Prusia
243-2870 r CO Grand Junction
692-2796 r PA King of Prusia
352-8563 t CO Greeley

(304)

345-7614 r WV Charleston
345-7140 + WV Charleston
345-6471 t WV Charleston
622-6827 * WV Clarksburg
529-9324 r WV Huntington
523-2802 t WV Huntington
292-0104 t WV Morgantown
292-0492 + WV Morgantown
296-0365 r WV Morgantown
485-3465 r WV Parkersburg
233-7732 t WV Wheeling
233-4096 r WV Wheeling

(305)

753-6571 r FL Ft Lauderdale
524-5304 + FL Ft Lauderdale
764-4505 t FL Ft Lauderdale
558-9124 r FL Miami
372-1355 + FL Miami
372-0230 t FL Miami
941-5445 t FL Pompano Beach

(306)

235-8611 r WY Casper
265-8807 + WY Casper
265-5167 t WY Casper
635-0151 r WY Cheyenne
638-4421 t WY Cheyenne
637-3958 + WY Cheyenne
745-8767 r WY Laramie
721-5878 t WY Laramie

(308)

381-2049 * NE Grand Island
(309)
827-7000 t IL Bloomington
662-8045 r IL Bloomington
828-2573 + IL Bloomington
674-9101 r IL Peoria
637-8570 t IL Peoria
637-8582 + IL Peoria
(312)
454-9538 r IL Chicago
938-8725 + IL Chicago
938-0600 t IL Chicago

(313)

665-2900 + MI Ann Arbor
930-0456 r MI Ann Arbor
996-5885 t MI Ann Arbor
964-2988 t MI Detroit
963-2274 + MI Detroit
964-0610 r MI Detroit
695-6610 r MI Flint
235-5477 + MI Flint
235-8517 t MI Flint
332-5120 t MI Pontiac
982-8364 t MI Port Huron
350-9582 r MI Southfield
827-4710 t MI Southfield
575-9152 t MI Warren
558-8460 + MI Warren
326-4210 * MI Wayne
(314)
449-4404 t MO Columbia

449-0729 r MO Columbia	724-9844 r GA Augusta	(414)	683-3773 r OR Eugene
634-8436 + MO Jefferson City	724-2752 t GA Augusta	734-7188 r WI Appleton	386-4405 t OR Hood River
634-4025 r MO Jefferson City	724-4494 + GA Augusta	432-2815 t WI Green Bay	882-6282 * OR Klamath Falls
634-5178 t MO Jefferson City	324-6882 r GA Columbus	432-0346 + WI Green Bay	779-6343 t OR Medford
421-4990 t MO St Louis	323-8931 + GA Columbus	435-3523 r WI Green Bay	776-4820 r OR Medford
421-0381 + MO St Louis	571-0556 t GA Columbus	552-9242 t WI Kenosha	773-7601 + OR Medford
725-7573 r MO St Louis	532-9880 * GA Gainesville	271-3914 t WI Milwaukee	297-2867 r OR Portland
(315)	234-1428 t GA Rome	258-6223 r WI Milwaukee	241-0496 + OR Portland
479-5423 r NY Syracuse	(405)	278-8007 + WI Milwaukee	295-3028 t OR Portland
479-5445 + NY Syracuse	353-4091 r OK Lawton	731-0620 t WI Neenah	378-1660 + OR Salem
472-5583 t NY Syracuse	353-0225 + OK Lawton	731-1560 + WI Neenah	378-7712 t OR Salem
797-0920 t NY Utica	353-0333 t OK Lawton	632-6166 t WI Racine	(504)
797-0228 + NY Utica	324-0187 r OK Oklahoma City	632-9188 r WI Racine	383-9634 r LA Baton Rouge
(316)	232-9513 + OK Oklahoma City	632-2174 + WI Racine	343-0753 t LA Baton Rouge
265-0727 r KS Wichita	232-4546 t OK Oklahoma City	452-3995 t WI Sheboygan	343-0771 + LA Baton Rouge
262-5669 t KS Wichita	372-8856 r OK Stillwater	452-7083 r WI Sheboygan	241-7260 r LA New Orleans
262-7961 + KS Wichita	743-1447 + OK Stillwater	334-2206 t WI West Bend	522-3967 + LA New Orleans
(317)	624-1112 t OK Stillwater	(415)	524-4094 t LA New Orleans
266-0962 r IN Indianapolis	(406)	827-3960 t CA Concord	(505)
299-6766 + IN Indianapolis	248-6373 + MT Billings	674-0127 + CA Concord	255-0108 r NM Albuquerque
299-0024 t IN Indianapolis	248-3338 r MT Billings	490-2050 t CA Fremont	242-1742 + NM Albuquerque
452-4744 r IN Kokomo	245-7649 t MT Billings	881-1382 t CA Hayward	243-4479 t NM Albuquerque
452-0073 + IN Kokomo	771-0067 t MT Great Falls	834-3194 + CA Oakland	525-3311 r NM Las Cruces
455-2460 t IN Kokomo	443-0527 + MT Helena	268-3782 r CA Oakland	526-9191 t NM Las Cruces
742-8485 r IN Lafayette	443-0000 t MT Helena	836-4911 t CA Oakland	662-5546 r NM Santa Fe
742-6000 t IN Lafayette	721-5900 t MT Missoula	856-0484 + CA Palo Alto	473-3403 t NM Santa Fe
289-0389 r IN Muncie	721-0453 r MT Missoula	856-9995 t CA Palo Alto	(507)
282-6418 t IN Muncie	543-5575 + MT Missoula	724-2400 t CA Pinole	388-3780 t MN Mankato
(318)	(407)	724-2225 + CA Pinole	285-1686 r MN Rochester
445-1053 t LA Alexandria	488-1788 r FL Boca Raton	734-8801 r CA Pleasanton	282-5917 t MN Rochester
473-4107 r LA Alexandria	338-3701 t FL Boca Raton	595-8870 + CA San Carlos	282-0253 + MN Rochester
235-8785 r LA Lafayette	267-0800 t FL Cocoa Beach	591-0726 t CA San Carlos	(508)
234-8451 + LA Lafayette	466-4566 * FL Fort Pierce	986-4292 r CA San Francisco	686-8199 r MA Andover
233-0002 t LA Lafayette	242-8247 t FL Melbourne	956-5777 t CA San Francisco	941-0110 r MA Brockton
436-9419 r LA Lake Charles	639-6090 r FL Melbourne	788-0823 + CA San Francisco	580-0721 t MA Brockton
436-0518 t LA Lake Charles	298-8201 r FL Orlando	377-1985 r CA San Mateo	677-4477 t MA Fall River
322-9826 + LA Monroe	422-4088 t FL Orlando	472-2550 + CA San Rafael	543-1535 r MA Foxboro
325-6215 r LA Monroe	422-8858 + FL Orlando	472-5360 t CA San Rafael	879-6798 t MA Framingham
387-6330 t LA Monroe	833-0382 r FL W Palm Beach	829-6705 t CA San Ramon	820-9241 r MA Framingham
424-2255 + LA Shreveport	833-6691 t FL W Palm Beach	256-8050 r CA Walnut Creek	975-2273 t MA Lawrence
221-5833 t LA Shreveport	655-2993 + FL W Palm Beach	(417)	452-1607 r MA Lowell
221-5124 r LA Shreveport	(408)	866-0122 r MO Springfield	937-5214 t MA Lowell
(319)	646-9092 t CA Monterey	864-4945 + MO Springfield	999-2915 t MA New Bedford
752-2516 * IA Burlington	646-1506 r CA Monterey	864-4814 t MO Springfield	999-9667 + MA New Bedford
364-0911 t IA Cedar Rapids	646-5122 + CA Monterey	(419)	990-1007 r MA New Bedford
362-2764 + IA Cedar Rapids	443-4940 t CA Salinas	353-2084 r OH Bowling Green	744-1559 t MA Salem
363-6965 r IA Cedar Rapids	443-8791 + CA Salinas	526-0686 t OH Mansfield	457-9390 + MA Woods Hole
324-8902 + IA Davenport	866-5784 r CA San Jose	747-7274 r OH Mansfield	540-7500 t MA Woods Hole
324-2445 t IA Davenport	294-9119 t CA San Jose	627-0050 t OH Sandusky	420-0973 r MA Woods Hole
556-0783 t IA Dubuque	286-6340 + CA San Jose	255-7881 t OH Toledo	792-1785 + MA Worcester
556-8702 r IA Dubuque	429-9192 + CA Santa Cruz	255-9922 r OH Toledo	797-3558 r MA Worcester
354-0676 + IA Iowa City	429-6937 t CA Santa Cruz	255-1906 + OH Toledo	755-4740 t MA Worcester
351-1421 t IA Iowa City	748-1391 r CA Sunnyvale	(501)	(509)
354-1033 r IA Iowa City	(409)	422-0210 * AR Fayetteville	235-6268 r WA Cheney
322-4949 r IL Moline	832-0180 r TX Beaumont	442-4654 r AR Fayetteville	332-0172 * WA Pullman
232-4776 r IA Waterloo	822-5287 r TX Bryan	782-2852 t AR Fort Smith	334-4154 r WA Pullman
232-0195 + IA Waterloo	822-0159 t TX Bryan	785-5544 r AR Fort Smith	943-0649 t WA Richland
232-5441 t IA Waterloo	765-7298 + TX Galveston	623-3159 * AR Hot Springs	838-9065 + WA Spokane
(401)	762-4382 t TX Galveston	374-2861 + AR Little Rock	624-7128 r WA Spokane
294-2415 r RI Kingston	938-0914 r TX Galveston	372-0748 r AR Little Rock	455-4071 t WA Spokane
295-7100 * RI North Kingston	722-3720 t TX Nederland	372-4616 t AR Little Rock	663-6227 t WA Wenatchee
732-6460 r RI Providence	727-4090 + TX Nederland	772-6181 * AR Texarkana	663-9180 + WA Wenatchee
831-3990 + RI Providence	(410)	(502)	575-1060 t WA Yakima
751-7912 t RI Providence	466-4566 * RI Woonsocket	782-6380 + KY Bowling Green	(512)
(402)	(412)	782-7941 t KY Bowling Green	929-3622 + TX Austin
475-3839 + NE Lincoln	836-4771 * PA Greensburg	875-1942 + KY Frankfort	928-1130 t TX Austin
477-1830 r NE Lincoln	856-1330 * PA Monroeville	875-4654 t KY Frankfort	480-9947 r TX Austin
475-4964 t NE Lincoln	471-6430 + PA Pittsburgh	429-6280 r KY Louisville	542-0367 t TX Brownsville
333-0242 r NE Omaha	931-2550 r PA Pittsburgh	583-1006 + KY Louisville	544-7073 + TX Brownsville
346-6419 + NE Omaha	288-9950 t PA Pittsburgh	589-5580 t KY Louisville	884-9030 t TX Corpus Christi
341-7733 t NE Omaha	(413)	686-8107 t KY Owensboro	884-6946 + TX Corpus Christi
(404)	586-0510 t MA Northampton	(503)	888-8294 r TX Corpus Christi
548-6395 r GA Athens	586-8666 r MA Northampton	754-0559 + OR Corvallis	725-9322 r TX Laredo
548-5590 t GA Athens	499-7741 t MA Pittsville	754-9273 t OR Corvallis	724-1791 t TX Laredo
523-0834 t GA Atlanta	781-3811 t MA Springfield	754-9608 r OR Corvallis	686-5360 t TX McAllen
396-1167 r GA Atlanta	737-9285 + MA Springfield	342-6626 + OR Eugene	686-2452 + TX McAllen
584-0212 + GA Atlanta	739-3805 r MA Springfield	683-1460 t OR Eugene	225-3444 + TX San Antonio

225-8004 t TX San Antonio	431-7592 + NH Portsmouth	385-2032 r MI Kalamazoo	355-2910 * IL Naperville (712)
271-0752 r TX San Antonio	433-8686 r NH Portsmouth	345-3088 t MI Kalamazoo	252-0206 r IA Sioux City
392-3401 r TX San Marcos	643-6517 r VT White River	345-3122 + MI Kalamazoo	255-1545 t IA Sioux City (713)
572-3197 * TX Victoria (513)	(605)	739-2348 r MI Muskegon	541-9382 r TX Houston
241-8008 + OH Cincinnati	224-2257 + SD Pierre	726-5723 t MI Muskegon	227-8208 + TX Houston
761-6440 r OH Cincinnati	224-0481 t SD Pierre	929-9873 r MI Traverse City	227-1018 t TX Houston (714)
579-0390 t OH Cincinnati	348-2621 t SD Rapid City	946-2121 t MI Traverse City (617)	635-6473 r CA Anaheim
427-1006 r OH Dayton	342-0680 r SD Rapid City	292-0662 t MA Boston	824-8976 + CA Colton
461-0755 + OH Dayton	336-6438 + SD Sioux Falls	890-4776 r MA Boston	824-9000 t CA Colton
461-5254 t OH Dayton	336-8593 t SD Sioux Falls	574-9244 + MA Boston	895-1207 + CA Garden Grove
863-4116 t OH Hamilton	338-6088 r SD Sioux Falls	542-2960 r MA Boston Center	898-9820 t CA Garden Grove
325-6466 r OH Springfield	(606)	326-4064 * MA Dedham	476-9376 r CA Irvine
324-1520 t OH Springfield (515)	269-3802 r KY Lexington	863-1745 + MA Lexington	622-1224 r CA Pomona
233-3460 r IA Ames	233-0312 t KY Lexington	863-1550 t MA Lexington	626-1284 t CA Pomona
233-6300 t IA Ames	233-7217 + KY Lexington	(618)	881-3385 r CA San Bernardino
233-2603 + IA Ames	(607)	277-5483 * IL Bellevue (619)	550-4625 + CA Santa Ana
243-2803 r IA Des Moines	724-2986 r NY Binghampton	747-5803 r CA Escondido	558-7078 t CA Santa Ana (715)
288-6202 + IA Des Moines	772-6642 t NY Binghampton	741-9536 + CA Escondido	834-2576 r WI Eau Claire
288-4403 t IA Des Moines (516)	772-9526 + NY Binghampton	741-7756 t CA Escondido	836-9295 t WI Eau Claire
243-1105 + NY Deer Park	257-7526 r NY Ithaca	430-0613 t CA Oceanside	836-0097 + WI Eau Claire
667-5566 t NY Deer Park	272-9980 + NY Ithaca	951-2615 * CA Palm Springs	845-9589 t WI Wausau
485-3380 + NY Hempstead	277-2142 t NY Ithaca	573-0224 r CA San Diego	842-5069 r WI Wausau (716)
292-3800 t NY Hempstead	(608)	233-0233 t CA San Diego	847-1825 + NY Buffalo
753-2030 r NY Melville (517)	362-5287 t WI Beloit	231-1703 + CA San Diego	884-5232 r NY Buffalo
782-8111 t MI Jackson	784-0560 t WI La Crosse	258-3887 r ND Bismarck	847-1440 t NY Buffalo
782-0523 r MI Jackson	257-8472 + WI Madison	235-7717 t ND Fargo	282-1462 t NY Niagara Falls
487-0833 r MI Lansing	257-4955 r WI Madison	293-0443 r ND Fargo	282-3284 + NY Niagara Falls
484-0062 t MI Lansing	257-5010 t WI Madison	235-9069 + ND Fargo	372-0583 r NY Olean
484-6301 + MI Lansing	(609)	775-7813 t ND Grand Forks	454-5730 + NY Rochester
832-2511 r MI Midland	645-6866 r NJ Atlantic City	663-6339 + ND Mandan	454-1020 t NY Rochester
832-7068 t MI Midland	344-8571 + NJ Atlantic City	663-2256 t ND Mandan	232-7526 r NY Rochester (717)
772-9447 r MI Mt Pleasant	348-0561 t NJ Atlantic City	(701)	249-9311 t PA Carlisle
799-3190 + MI Saginaw	273-1411 r NJ Cherry Hill	258-3887 r ND Bismarck	275-0151 r PA Danville
790-5166 t MI Saginaw (518)	596-1500 t NJ Marlton	235-7717 t ND Fargo	271-0102 t PA Danville
465-8444 t NY Albany	596-8659 + NJ Marlton	293-0443 r ND Fargo	236-6882 t PA Harrisburg
383-4435 r NY Albany	663-9297 t NJ Merchantville	235-9069 + ND Fargo	236-2007 + PA Harrisburg
465-8632 + NY Albany	665-6860 + NJ Merchantville	775-7813 t ND Grand Forks	564-9577 r PA Harrisburg
562-1890 t NY Pittsburgh (601)	936-0231 + NJ Princeton	663-6339 + ND Mandan	295-5405 t PA Lancaster
436-6719 r MS Biloxi	799-5587 t NJ Princeton	663-2256 t ND Mandan	293-0882 r PA Lancaster
863-0024 t MS Gulfport	989-8847 t NJ Trenton	(702)	961-5321 t PA Scranton
264-0815 * MS Hattiesburg	989-7127 + NJ Trenton	737-5466 + NV Las Vegas	969-9760 r PA Scranton
584-7229 r MS Hattiesburg	882-8835 r NJ Trenton	737-6861 t NV Las Vegas	961-5480 + PA Scranton
372-7869 r MS Jackson	(612)	873-0434 r NV Las Vegas	829-3108 t PA Wilkes-Barre
969-0152 + MS Jackson	341-2459 t MN Minneapolis	322-1200 r NV Reno	823-1640 r PA Wilkes-Barre
969-0036 t MS Jackson	858-3539 r MN Minneapolis	827-5290 + NV Reno	824-8209 + PA Wilkes-Barre
482-2210 t MS Meridian	338-1661 + MN Minneapolis	827-6900 t NV Reno	494-1796 t PA Williamsport
236-1658 r MS Oxford	253-4248 r MN St Cloud	(703)	322-4390 r PA Williamsport
324-2155 t MS Starkville	253-2064 t MN St Cloud	845-0405 r VA Alexandria	843-5141 r PA York
323-1043 r MS Starkville (602)	(614)	552-9181 t VA Blacksburg	846-6550 t PA York
773-0588 * AZ Flagstaff	592-3078 r OH Athens	552-3110 r VA Blacksburg	843-0039 + PA York (719)
937-0596 r AZ Glendale	263-2458 r OH Columbus	962-2217 t VA Covington	380-0022 r CO ColoradoSprngs
893-6670 r AZ Phoenix	461-9044 + OH Columbus	371-0188 t VA Fredericksburg	635-5361 t CO ColoradoSprngs
256-6955 + AZ Phoenix	463-9340 t OH Columbus	433-4868 r VA Harrisonburg	578-0950 + CO ColoradoSprngs
254-0244 t AZ Phoenix	(615)	434-0374 + VA Harrisonburg	542-4053 t CO Pueblo
747-0107 t AZ Tucson	968-1130 t TN Bristol	434-7121 t VA Harrisonburg	545-3411 r CO Pueblo (800)
747-9395 + AZ Tucson	265-7929 + TN Bristol	481-6807 + VA Herndon	272-5299 * HI Other islands (801)
571-0275 r AZ Tucson (603)	756-1161 t TN Chattanooga	435-1800 t VA Herndon	752-3421 * UT Logan
225-8710 + NH Concord	622-7823 r TN Chattanooga	494-0836 * VA Occoquan	753-7865 r UT Logan
228-4499 r NH Concord	265-7929 + TN Chattanooga	344-2404 + VA Roanoke	627-1630 t UT Ogden
224-1024 t NH Concord	552-0032 t TN Clarksville	344-2036 t VA Roanoke	627-1640 + UT Ogden
868-2924 t NH Durham	282-1104 r TN Johnson City	989-8952 r VA Roanoke	375-2084 + UT Provo
643-6517 r NH Hanover	282-6645 t TN Johnson City	(704)	373-0542 t UT Provo
647-7310 r NH Manchester	690-1957 r TN Knoxville	252-9134 t NC Asheville	375-6005 r UT Provo
625-8088 + NH Manchester	523-5500 t TN Knoxville	254-1749 r NC Asheville	359-0578 + UT Salt Lake City
627-8725 t NH Manchester	521-5072 + TN Knoxville	252-0133 + NC Asheville	359-0149 t UT Salt Lake City
882-4588 r NH Nashua	255-2608 + TN Nashville	336-6970 r NC Charlotte	484-5011 r UT Salt Lake City (802)
880-3901 + NH Nashua	360-2100 r TN Nashville	333-6204 + NC Charlotte	658-2410 r VT Burlington
880-6241 t NH Nashua	244-3702 t TN Nashville	332-3131 t NC Charlotte	
431-2302 t NH Portsmouth	481-3590 t TN Oak Ridge	865-4708 t NC Gastonia	
	483-7875 r TN Oak Ridge	(707)	
	(616)	444-3091 t CA Eureka	
	968-0929 t MI Battle Creek	579-2642 r CA Santa Rosa	
	963-6881 r MI Battle Creek	578-4447 t CA Santa Rosa	
	968-9851 + MI Battle Creek	578-1055 + CA Santa Rosa	
	774-0966 t MI Grand Rapids	643-3592 r CA Vallejo	
	361-9363 r MI Grand Rapids	(708)	
	774-0131 + MI Grand Rapids	255-6465 * IL Arlington Hts	
		820-2005 r IL Aurora	
		896-3363 + IL Aurora	
		896-0620 t IL Aurora	
		825-5070 r IL Des Plaines	
		362-7838 * IL Libertyville	

864-0808 t VT Burlington	337-5401 r FL Fort Myers	732-5581 r FL Ocala	944-7621 t TX San Angelo
229-4966 t VT Montpelier	687-0666 + FL Lakeland	763-8377 * FL Panama City	(916)
775-1676 t VT Rutland	686-7155 r FL Lakeland	476-7215 r FL Pensacola	894-6882 t CA Chico
775-5999 r VT Rutland	683-5461 t FL Lakeland	432-1335 t FL Pensacola	753-4387 t CA Davis
295-7631 t VT WhiteRiver Jnt (803)	263-3033 t FL Naples	434-2103 + FL Pensacola	742-4114 r CA Marysville
577-4710 + SC Charleston	261-7227 r FL Naples	386-3828 r FL Tallahassee	243-0690 * CA Redding
722-4303 t SC Charleston	743-4602 r FL Port Charlotte	681-1902 t FL Tallahassee	224-0920 r CA Redding
797-8405 r SC Charleston	365-7311 r FL Sarasota	561-8830 + FL Tallahassee	448-6262 t CA Sacramento
653-8702 r SC Clemson	925-1499 + FL Sarasota	(907)	443-7434 + CA Sacramento
254-0695 t SC Columbia	923-4563 t FL Sarasota	276-0453 t AK Anchorage	362-4103 r CA Sacramento
252-0328 + SC Columbia	323-4026 t FL St Petersburg	852-2425 t AK Barrow	(918)
252-6806 r SC Columbia	327-1163 + FL St Petersburg	543-2411 t AK Bethel	336-0461 r OK Bartlesville
271-0231 + SC Greenville	223-5859 + FL Tampa	532-2371 t AK Cold Bay	336-3675 t OK Bartlesville
233-3486 t SC Greenville	286-2504 r FL Tampa	424-3744 t AK Cordova	663-2606 r OK Tulsa
242-3474 r SC Greenville	224-9920 t FL Tampa	659-2777 t AK Deadhorse	584-3247 t OK Tulsa
585-1637 t SC Spartanburg	897-3421 * FL Valparaiso	895-5070 t AK Delta Junction	587-2774 + OK Tulsa
583-1412 r SC Spartanburg (804)	(814)	842-2688 t AK Dillingham	(919)
977-8818 r VA Charlottesville	949-0310 t PA Altoona	456-3282 t AK Fairbanks	323-4148 + NC Fayetteville
977-5330 t VA Charlottesville	453-3793 + PA Erie	822-5231 t AK Glennallen	864-6802 r NC Fayetteville
838-7119 r VA Hampton	454-4863 r PA Erie	235-5239 t AK Homer	323-8165 t NC Fayetteville
845-0010 t VA Lynchburg	536-3630 + PA Johnstown	571-1364 t AK Iliamna	275-1251 + NC Greensboro
847-6558 r VA Lynchburg	535-7576 t PA Johnstown	789-7009 t AK Juneau	273-6714 r NC Greensboro
596-6600 t VA Newport News	234-2820 r PA State College	225-1871 t AK Ketchikan	273-2851 t NC Greensboro
596-2710 + VA Newport News	231-1510 t PA State College	246-3049 t AK King Salmon	830-4931 r NC Greenville
625-1186 t VA Norfolk	(815)	486-4061 t AK Kodiak	889-7494 t NC High Point
623-1480 r VA Norfolk	758-2042 r IL DeKalb	442-2602 t AK Kotzebue	838-9034 t NC NthWilkesboro
625-2408 + VA Norfolk	758-2623 t IL DeKalb	524-3256 t AK McGrath	549-8175 r NC Raleigh
965-0871 r VA Richmond	758-5046 + IL DeKalb	443-2256 t AK Nome	834-8254 * NC Raleigh
788-9902 t VA Richmond	726-8731 + IL Joilet	778-2301 t AK Northway	549-8139 t NC Res Tri Park
343-4140 + VA Richmond (805)	726-0070 t IL Joilet	745-0200 t AK Palmer	541-9096 + NC Res Tri Park
321-0112 r CA Bakersfield	722-8517 r IL Joilet	659-2777 t AK Prudhoe Bay	823-0578 t NC Tarboro
327-8146 t CA Bakersfield	965-0400 t IL Rockford	224-3126 t AK Seward	763-8313 t NC Wilmington
327-5321 + CA Bakersfield	965-0696 + IL Rockford	747-5887 t AK Sitka	343-0252 r NC Wilmington
723-6729 r CA Lancaster	962-5433 r IL Rockford	262-1990 t AK Soldotna	251-8900 + NC Wilmington
486-8699 r CA Oxnard	(816)	546-2320 t AK St Paul	725-2126 t NC Winston-Salem
544-1979 r CA San Luis Obispo	842-3009 r MO Kansas City	733-2227 t AK Talkeetna	724-1280 r NC Winston-Salem
966-7868 r CA Santa Barbara	472-1430 + MO Kansas City	366-7167 # AK Tanana	777-0312 + NC Winston-Salem
564-1158 + CA Santa Barbara	221-9900 t MO Kansas City	835-4987 t AK Valdez	
682-5361 t CA Santa Barbara	279-4797 t MO St Joseph	472-2467 t AK Whittier	
925-7039 * CA Santa Maria	233-6273 r MO St Joseph	784-3453 t AK Yakutat	
928-8857 r CA Santa Maria	(817)	(912)	
495-3588 t CA Thousand Oaks	381-1897 * TX Denton	888-3011 t GA Albany	
656-6760 t CA Ventura	332-6794 + TX Fort Worth	431-9384 + GA Albany	
650-1070 + CA Ventura (806)	548-0171 r TX Fort Worth	439-1918 r GA Albany	
373-1833 + TX Amarillo	332-4307 t TX Fort Worth	746-0568 r GA Macon	
358-6277 r TX Amarillo	773-9723 t TX Temple	741-2108 + GA Macon	
373-0458 t TX Amarillo	770-0560 r TX Temple	743-8844 t GA Macon	
747-4121 t TX Lubbock	776-9533 r TX Waco	236-2875 + GA Savannah	
744-6291 r TX Lubbock (808)	752-9743 t TX Waco	966-5775 r GA Savannah	
839-7234 r HI Honolulu	322-3774 t TX Wichita Falls	236-2605 t GA Savannah	
528-0200 * HI Oahu (809)	(818)	(913)	
891-5070 t PR Aguadilla	999-9068 r CA Canoga Park	749-5901 r KS Lawrence	
878-4200 t PR Arecibo	444-1173 r CA El Monte	843-8124 t KS Lawrence	
860-5690 t PR Fajardo	246-3886 + CA Glendale	651-0015 * KS Leavenworth	
850-8015 t PR Humacao	507-0909 t CA Glendale	539-8118 r KS Manhattan	
854-6500 t PR Manati	584-1194 r CA Pasadena	537-0948 t KS Manhattan	
831-7032 t PR Mayaguez	376-1004 r CA Van Nys	825-7900 t KS Salina	
841-8088 t PR Ponce	915-5151 t CA West Covina	233-2880 t KS Topeka	
792-3030 t PR San Juan (812)	887-3160 t CA Woodland Hills	233-4660 + KS Topeka	
330-0005 r IN Bloomington	348-7141 + CA Woodland Hills	233-2884 r KS Topeka	
332-1344 t IN Bloomington	(901)	(914)	
424-7693 t IN Evansville	527-5175 + TN Memphis	473-3200 + NY Poughkeepsie	
428-2522 + IN Evansville	452-7111 r TN Memphis	473-2240 t NY Poughkeepsie	
234-4119 + IN Terre Haute	521-0215 t TN Memphis	446-2950 r NY West Point	
234-2484 r IN Terre Haute	(904)	747-2892 r NY White Plains	
232-5329 t IN Terre Haute (813)	255-2629 t FL Daytona Beach	328-9199 t NY White Plains	
275-4153 + FL Cape Coral	258-7007 r FL Daytona Beach	682-3505 + NY White Plains	
275-7924 t FL Cape Coral	374-8700 r FL Gainesville	(915)	
441-9948 r FL Clearwater	338-1700 + FL Gainesville	676-9151 t TX Abilene	
	338-0220 t FL Gainesville	778-8196 r TX El Paso	
	257-4770 * FL Holly Hill	541-1931 + TX El Paso	
	353-1818 t FL Jacksonville	532-7907 t TX El Paso	
	791-9201 + FL Jacksonville	697-0462 r TX Midland	
	355-6102 r FL Jacksonville	261-9811 t TX Midland	
	787-0799 * FL Leesburg	561-8597 + TX Midland	
	351-3790 t FL Ocala	949-1231 + TX San Angelo	
		655-3608 r TX San Angelo	

ORDER FORM Subscriber #_____

Name: _____

Street: _____

City/St/Zip: _____

Visa/MC#: _____

Exp Date: _____ Signature: _____
Dec '90-Prices subject to change without notice

TO ORDER:

- Fill in subscriber# / address (above)
(new subscribers write new for subscriber #)
- Check [✓] each item you want to order
- Add up items/shipping/handling (fill in below)
(shipping fee is often less than the max listed)
- Send check/money order or charge it (above)
- Charge orders may call 608-222-4432
- or Mail to: COMAL Users Group, U.S.A., Ltd.
5501 Groveland Terrace, Madison, WI 53716

SUBSCRIPTIONS:

Renewals start with the issue where you left off.

- <= How many issues? \$4 per issue (max 2 past current)
(Canada/APO add \$1 per issue, 1st Class)
- <= How many disks? \$9 per disk (max 2 past current)
(C64 only before 25) Disk Type: _____

- \$4.95 each for backissues; circle ones wanted:
1 2 4 5 6 7 8 9 10 11 12 13 14
15 16 17 18 19 20/21 22 23 24 25 26
- \$9.95 each, C64 COMAL Today Disks; circle to order:
1/2 3/4 5 6 7 8 9 10 11 12 13 14
15 16 17 18 19 20/21 22 23 24 25 26

Notice: All orders must be prepaid in US dollars. Minimum order \$10. Minimum shipping is \$3 (does not apply to subscription or Fish Disk only orders); Canada, APO & 1st Class add \$1 more per newsletter and book. Newsletter is published as time permits (no set schedule); size and format varies. Cancelled subscriptions receive no money back. Orders accepted from Canada and USA only. Allow 2 weeks for checks to clear. \$15 charge for checks not honored. Prepaid PO's from companies accepted only if no conditions of sale apply. Wisconsin residents add 5% sales tax, plus your county tax, if any.

ENTER TOTALS HERE:

Item Total (\$10 minimum):\$_____

Ship Total (\$3 minimum):\$_____

Wisconsin (& county) tax:\$_____

Grand Total enclosed:\$_____

BACKISSUES - COMAL Today

Price per issue: \$4.95 (no extra shipping other than minimum)
(circle numbers on the left; matching disks are also available)

- 1 - Original first issues still available (not reprints). Only a few copies left.
- 2 - PRINT FILE versus WRITE FILE; Recursion; Stack overflow; Functions/Parameters
- 3 - Sold Out!
- 4 - Graphics screen dump in COMAL; spirolateral; Dodge 'em; Music; arrays /parameters
- Issues 1-4** Spiral bound: (see books: COMAL Yesterday)
- 5 - How COMAL statements are stored; strings; Target; Inventory; Pitfall Harry
- 6 - Memory maps; 2.0 & 0.14 compared; Shadow letters; Draw molecule; Function keys
- 7 - COMAL Standards meeting; Book reviews; Function keys; External procs
- 8 - Sprites; Soundex; Forest; Sound; Fun print; Recursion; Font sprites; Depreciation
- 9 - Modems; Ascii codes; TRAP; Function keys; Metamorphose sprite; Bitmaps; Sizzle; C64 cart internal structure & token table; Rod the Roadman; Strings; Icon Editor
- 10- Sorts; Font Editor; Draw; Walker; Phone Database; Missing Letters; Designer; Compressed bitmaps; Meta; Compare disk files; Easy Curves
- 11- 2.0 keywords chart; Disk directories; Pop Over; Graphics Editor System; Ram disk
- 12- Cart package keywords chart; Rabbit; 3D Fractals; Cart schematics; 2 column printing; Pic Finder; Benchmarks; Free form database; Transfer programs from 0.14 to 2.0; Kelly's Beach
- Index to issues 1-12**, 4,848 entries, 56 pgs; (see books)
- 13- Superchip keywords chart; Sprites; Wheel of Fortune; Sets; Benchmarks; BASIC into COMAL; Encryption; C128 package; Kastle
- 14- Outliner; Listerine; Scope; Stacks; CASE statement; Calendar; Multi-directory; Modems
- 15- Over 60 0.14 Proc/Func listed!; Dr Who database; Program construction; Wheel of Fortune
- 16- Smart file reader; Text input window; Magic squares; Easy instructions; Read & Run; Learn subtraction; NIM; Tiny directory; Sorts
- 17- COMAL Kernel in full; Ram expander; Sprites; Calculate PI; 0.14 graphic/sprite chart
- 18- Reversi; Poetry; Hammurabi; Puzzle; Zip Zone; Fractal geometry / Mandelbrot; stacks
- 19- Power Driver keywords chart; Sample book pages; Coloring book; Rotating 3D
- 20/21 - Files; Black box; Best of 1-19
- 22- Walking sprites; Animals; Maze; Music
- 23- Message board; Programming; Graphing
- 24- Common COMAL; Picture Package; Chaos
- 25- Story of COMAL by Borge Christensen
- 26- PLINK Info; Converter; IBM PC COMAL
- Index to issues 13-26**, (see books)

SYSTEMS

ORDER FORM

SYSTEMS

C64 - Disk Loaded

- Power Driver Complete^{db} - includes Power Box, Starter Kit, keyboard overlay. \$49.95 + \$6 shipping
- Add \$9 for Doc Box^x binder/slipcase

- Power Driver interpreter and 20 lesson tutorial (with turtle tutor) on disk. \$9.95

- Power Box^{db} - includes Power Driver interpreter and compiler, 3 utility disks, a toolbox of about 250 procedures and functions and Power Driver note pages. \$29.95 + \$4 ship

- Starter Kit - 12 issues of *COMAL Today*, 56 page index, 2 books, 5 disks! Over 1,000 pgs! \$29.95 + \$4 ship

- C64 Keyboard Overlay: excellent condensed command reference. \$3.95 + \$1 shipping

- Doc Box^x binder/slipcase option ... \$9

C64 - Cartridge

- COMAL 2.0 Cart Complete*
The C64 cartridge plus all the options below (except Superchip) ... Tutorials, references, packages, applications ... \$179.95 + \$7 ship

- COMAL 2.0 Cart Deluxe*
64K cartridge; 3 reference books: *2.0 Keywords*, *Cart Graphics & Sound*, *Common COMAL Reference* plus 4 demo disks. \$124.95 + \$3 ship

- COMAL 2.0 Cartridge*
64K cartridge with empty spot for up to 32K EPROM. Plain, no documentation. \$99.95

- Deluxe Option^{db}: three books: *2.0 Keywords*, *Cart Graphics & Sound*, *Common COMAL Reference*; 4 cart demo disks. \$29.95 + \$4 ship

- Packages Option^{db}: three books: *COMAL 2.0 Packages*, *Packages Library 1* (17 packages), *Packages Library 2* (24 packages); Superchip on Disk (9 packages). \$36.95 + \$4 ship

- Applications/Tutorial Option^{db}: *COMAL Collage*, *3 Programs In Detail*, *Graph Paper*. \$36.95 + \$4 ship

- Super Chip On Disk: adds about 100 commands to the cartridge. \$14.95

- Doc Box^x Binder/Slipcase Option ... \$9

IBM PC & compatibles (prices include shipping)
 Please specify disk type: _____

- Entry Level - UniComal IBM PC COMAL 3.02*
With compiler, graphics, reference, tutorial. \$235
- Standard Level - UniComal IBM PC COMAL 3.02*
With timer, mouse, graphics, reference, tutorial. \$435
- Extended Level - UniComal IBM PC COMAL 3.02*
Adds compiler & SCOM to Standard Level system \$635
- Developers Level - UniComal IBM PC COMAL 3.02*
Adds Assembler & C tools, BTrieve interface \$835

- Option: SCOM \$300 (serial communication)
- Option: Menus & Windows \$125
- Option: Common Comal Reference \$16.95 + 3 ship
- IBM Today Disk 25: \$9.95
- IBM Today Disk 26: \$9.95 (with COM-AND term program)
- IBM Special Series #1: \$9.95
- IBM Special Series #2: \$9.95
- Upgrades and Educational Licenses - please write!

Amiga (all models)

- AmigaCOMAL 2.0+ (including 2.04 update)
Full fast system, graphics, speech, exceptions, windows, screens, devices \$99.95 + \$3 ship (includes Today 25 programs)
- Compiler/Developer Option: \$34.95 + \$1 ship \$FREE...
FREE with AmigaComal now
- AmigaCOMAL 2.04 UPDATE disk \$9.95
- Today Disk 26: \$9.95 (with NComm term program)

Amiga - FISH DISKS

- FISH Disks, just \$1.95 each (include list of #'s)
- Fish Catalog Disk: \$1.95 (view from WorkBench or CLI)
(includes the contents of every Fish Disk, plus a quick reference file)
- Fish Disk Subscription (requires VISA or MC plus a standing order request - give us a starting disk #)

Amiga - Special Disk Collections

- Borge Christensen digitized pics disk: \$9.95
- Text Editors & Readers disk set: \$7.90 (2 disks: HED, JED, QED, AVI, Textra,... More, Less, MuchMore, Most,...)
- Archivers disk set: \$7.90 (2 disks: Lharc, Zoo, Zip,...)
- Modem disk set: \$7.90 (2 disks: Access, GT, JRComm,...)
- Font disk set: \$7.90 (2 disks: many different fonts)

COMAL Information Booklets

_____ < = How many? (10¢ each)

db = Doc Box pages

* = subject to customs/ship variations/availability

^x = while supplies last (out of print)

DISKS

ORDER FORM

DISKS

Disks are \$9.95 each (unless another price is shown). Unless the disk label specifically states that you may not give out copies, our disks may be copied or placed into local club disk libraries. Choose from the disks below:

- [] Amiga COMAL Demo Disk (\$2)
- [] Amiga Today Disk 26 (with modem term program)
- [] C64 Today Disk 26 (with modem term program)
- [] IBM Today Disk 25
- [] IBM Today Disk 26 (with modem term program)
- [] IBM Special Series Disk #1
- [] IBM Special Series Disk #2 (Test System)

- [] Beginning COMAL disk §
- [] Today INDEX disk § (2 disks count as 1)
- [] Games Disk #1 (0.14 & 2.0)
- [] Modem Disk (0.14 & 2.0)
- [] Article text files disk

Today Disks:

- [] Today Disk (one disk type—circle choices):
1 2 3 4 5 20&21 24 25 26
- [] Today Disk (double sided — circle choices):
6 7 8 9 10 11 12 13 14 15 16 17 18 19 22 23

0.14 / Power Driver Disks:

- [] Sid Siferlein disk set (3 disks count as 1)
- [] Data Base Disk 0.14
- [] New: Power Driver Tutorial Disk
- [] Auto Run Demo Disk
- [] Paradise Disk
- [] Best of COMAL
- [] Bricks Tutorial (2 disks count as 1)
- [] Utility Disk 1
- [] Slide Show disk (circle which): 1 2
- [] Spanish COMAL
- [] User Group 0.14 disks (circle numbers):
1 2 3 4 5 6 7 8 9 10 12

2.0 Disks:

- [] Robert Ross Technical Disk (\$2)
- [] Ajax School disk set (2 disks count as 1)
- [] Norquay School disk
- [] Data Base Disk 2.0
- [] Superchip Programs disk
- [] Read & Run
- [] Math & Science
- [] Typing disk (2 disks count as 1)
- [] Cart Demo (circle which): 1 2 3 4
- [] 2.0 user disks (circle choices): 11 13 14 15

§ = these disks assume you have the book

Note: Some disks may be supplied on the back of another disk. Disk format is C64 unless specified otherwise. We replace defective disks at no charge if you return the disk with a note explaining what is wrong with it. Some disks are being reduplicated and appropriately relabeled.

\$10.95 Sprite Pak

Two disk set. Huge collection of sprite images, sprite editors, viewers, and other sprite programs. For 0.14 and 2.0.

\$12.95 Font Pak

Three disk set. Collection of many different character sets (fonts) for use with 0.14 and 2.0 including special font editors!

\$14.95 Graphics Pak

Five disk set. Picture heaven. Includes Slide Show, Picture Compactor, Graphics Editor and lots of pictures (normal and compacted)

\$29.95 Sprite,Font & Graphics Pak

All ten disks mentioned above!

\$10.95 Guitar Disks

Three 0.14 disk set. Teaches guitar by playing songs while displaying the chords and words.

\$14.95 Cart Demo Disks

Four disks full of programs demonstrating the many features of the C64 2.0 cartridge.

\$10.95 Shareware Disks

Three disk set. Includes a full HazMat system (Hazardous Materials), an Expert System, Finger Print system, Traffic Calc, and a BBS program.

\$14.95 Superchip On Disk

All the commands of Super Chip (but not the Auto Start feature) disk loaded.

\$24.95 Super Chip Source Code

Full source code with minimal comments. Customize your own Super Chip. Add commands. Remove the ones you don't need.

\$14.95 2.0 User Group Disks

Four disks set for the C64 COMAL cart.

\$29.95 0.14 User Group Disks

Twelve disks (User Group disk 9 is a newsletter on disk system, double disk).

\$29.95 European Disk Set

Twelve 2.0 disk set. Find out what COMALites in Europe are doing.

BOOKS

ORDER FORM

BOOKS

\$2.95 C64 COMAL 2.0 Keywords^{db}

#4 best seller for Feb 88 lists all the keywords built into the cartridge (including all 11 built-in packages) in alphabetic order complete with syntax and example. +\$1 ship

\$16.95 Beginning COMAL^a

#8 best seller by Borge Christensen
333 pages - General Textbook

Beginners text book, elementary school level, written by the founder of COMAL. This book is an easy reading text. You should find Borge has a good writing style, with a definite European flair. +\$3 ship

Optional Matching Disk of programs. \$9.95

\$3.95 COMAL From A to Z^a

#1 all time best seller by Borge Christensen
64 pages - Mini 0.14 Reference book
Written by the founder of COMAL. +\$1 ship

\$3.95 COMAL Workbook^a

#1 best seller for Feb 88 by Gordon Shigley
69 pages - 0.14 Tutorial Workbook; Companion to the Tutorial Disk, great for beginners, full sized fill in the blank style. +\$1 ship

Tutorial Disk Option: \$9.95

\$3.95 Index - COMAL Today 1-12^a

#9 best seller by Kevin Quiggle
52 page, 4,848 entry index to COMAL Today. The back issues of COMAL Today are a treasure trove of COMAL information and programs! This index is your key. +\$1 ship

Index Disk Option: \$9.95

Coming Soon: INDEX - COMAL TODAY 13-26

\$16.95 Common COMAL Reference^{db}

#3 best seller for Feb 88 by Len Lindsay
238 page detailed cross reference to the COMAL implementations in the USA (formerly COMAL Cross Reference) Covers: C64 COMAL 2.0, C128 COMAL 2.0, CP/M COMAL 2.10, and UniComal IBM PC COMAL. +\$3 ship

\$19.95 COMAL Yesterday: Issues 1-4 of COMAL Today - Spiral bound +\$3 ship

\$12.95 Library of Funcs/Procs^{db}

#1 best seller Dec 87 by Kevin Quiggle, 80 pgs, over 100 0.14 procs/funcs, with disk. +\$1 ship

\$4.95 Cart Graphics & Sound^{db}

#6 best seller by Captain COMAL
64 pgs - 2.0 packages reference guide to all commands in the 11 built in cartridge packages. +\$1 ship

\$14.95 COMAL 2.0 Packages^{db}

#7 all time best seller by Jesse Knight
108 pgs with disk - package reference. How to write a package in Machine Code; includes C64 comsym & supermon. For advanced users. +\$2 ship

\$14.95 Package Library Vol 1^{db}

compiled by David Stidolph
76 pages with disk - package collection
17 packages ready to use, many with source code, plus the Smooth Scroll Editor! +\$1 ship

\$14.95 Package Library Vol 2^{db}

67 pages with disk - package collection
24 example packages ready to use, most with source code, plus Disassembler, Re-Linker, De-Linker, Package Maker, Package Lister, and more. (includes windows). +\$1 ship

\$14.95 COMAL Collage^{db}

by Frank and Melody Tymon
168 pages with disk, 2.0 programming guide, including graphics and sprites tutorial with many full sized example programs. +\$2 ship

\$12.95 3 Programs in Detail^{db}

82 pages with disk by Doug Bittinger
Three 2.0 application programs explained: Blackbook (name/address system), Home Accountant, and BBS. +\$1 ship

\$12.95 Graph Paper^{db}

52 pages with disk by Garrett Hughes
Function graphing system for COMAL 2.0. The program can't be LISTed. Includes a version for the Commodore Mouse. +\$1 ship

\$12.95 COMAL Quick /Utility 2 & 3^{db}

#2 best seller Dec 87 by Jesse Knight
20 pgs with 2 disks, fast loading COMAL 0.14, printer programs, utility programs. +\$1 ship

\$24.95 Intro to COMAL Advance Orders

Textbook for all COMALs by J William Leary and Len Lindsay. Aprox. 300 pages. +\$3 ship

Programs disk option: \$5.95 Check format needed:

Amiga IBM C64 (double disk)

[=] \$16.95 Foundations with COMAL^a sold out

db = Doc Box pages

* = subject to customs/ship variations/availability

□ = while supplies last (out of print)

Alphabetical listing, by program name, of Fish disks 1 through 390
Compiled from 'Contents' files using Transcript, Professional Page, and AmigaCOMAL

'Liner	285	AssignDev	079	Calc	103	CRobots	311	DM-Maps	240	FBM	334	GravityWars	084	InStiff	043
3D-Arm	047	AssignDev	111	Card	210	CRobot	311	Dmake	179	Fd	075	GravWars	105	InUDOS	103
3DTicToe	366	Assigned	066	CalcKey	235	CrossDOS	382	Dmake	246	Fedup	230	GravSim	223	InUface	366
60x80	157	Asteroids	122	Calculation	260	CrossDOS	240	Dme	059	Fenser	362	grep	004	InUSup	316
A-Render	099	AsteroidField	208	Calendar	032	CrossDOS	252	Dme	074	Fenser	245	Guardian	154	InUtionEd	377
A68k	110	ATO	245	Calendar	034	Csh	024	Dme	087	Fenser	298	Guru	239	IPC	290
A68k	186	ATPatch	061	Calendar	269	Csh	036	Dme	093	Fenser	305	GurusGuide	203	IPDevice	374
A68k	314	Area	289	Calls	160	Csh	041	Dme	113	ff	003	Gwin	322	IPo2C	105
ABASIC_PROGS	013	AudioTools	084	Car	123	Csh	055	Dme	134	FFT	239	IPoPrint	367	IRA	304
ABFemos	027	AudioTools	094	Car	336	Csh	085	Dme	153	File	231	H2I	387	Isam	203
ABridge	359	AutoEnquirer	089	CardFile	080	Csh	107	DME	168	FileBootBlock	191	H2J	239	ISpell	054
Access	098	AutoFacc	081	CardFile	090	Csh	145	DME	169	FileEncrypt	379	HackBench	096	ISpell	191
Accordion	260	AutoGraf	166	CardMaker	375	Csh	199	Dme	284	JARSG-Demo	086	HackLite	187	Itb	326
Acp	036	AutolconOpen	073	CardMaker	184	Csh	223	Dme	365	Field	257	Hack_1.0.1	007	JAR	242
Ada	154	AutolconOpen	087	Cards'ORama	186	Csh	309	DmeMacros	146	Field	307	Hack_1.0.1	008	Jask	177
Adapt	378	Automata	249	CaseEfi	346	Csh	331	Dmouse	145	Field	348	Hack_1.0.1.E	025	JazzMinesides	019
Add	073	AutoPoint	086	Castle	174	Csh2.04	048	Dmouse	160	Fleet	220	Hack_1.0.3.D	062	JazzBench	228
ADDR	168	AuxHandler	079	CBBS	241	Csh2.04.M	048	Dmouse	168	FleetMaster	288	HagenDemos	112	JeanIcons	137
ADDR	169	AvailMem	285	CBDump	326	CSquared	039	Dmouse	169	FleetMaster	361	halbwite	001	Jed	180
AddKickMem	105	Avi	162	cc	002	Ci	137	Dmouse	238	FleReq	085	HAMCU	204	Jed	297
Address	032	AvTrees	103	Cc	029	Ci	235	Dmouse	258	FleReq	204	Hames	328	Jive	046
ADL	091	Az	228	Cclib	260	CTags	197	Dnet	145	FleReq	242	HAMmm	118	Jobs	070
Adventure	082	Az	346	Ctype	286	Ctype	237	Dnet	220	FleRequest	292	HAMmm2	239	Journal	095
AdvSys	087	AZComm	171	Ccutils	286	Cucug	213	Dnet	294	FleRequester	035	HanPics	196	JoyLab	378
AegisDrawDemo	029	AztecArp	353	Cdecl	114	Cursor	347	Dos2split	304	FILES	168	HAMpily	052	joystick	005
AltTerm	170	AztecArp	376	CDecl	167	Curves	105	Doctor_A	268	FILEs	169	HAM DEMO	017	JPClock	204
AHDm	319	Back	284	clorth	003	CustReq	242	DogsWorld	212	FileSelect	173	Handler	038	JPClock	242
Ahost	040	BackDrop	216	Charon	198	CutAndPaste	057	DoRevision	325	FileSystems	329	HandShake	060	JDirUtil	287
AirFoil	071	Backgammon	028	Check	160	CutAndPaste	097	DoDev	113	FileTest	284	Handshake	172	Jsh	038
AirFoil	150	BadGammon	120	Check4Mem	242	CWDemo	238	DoError	366	FileWindow	336	Juggler	099	Juggler	047
AlarmingClock	229	Backup	098	CheckIFF	081	CWDemo	283	DOSHelper	040	Filezap	010	Hangman	299	Juggler	057
AlgoRhythms	356	Backup	168	CheckModem	046	Cycles	078	DosKwik	103	Filezap	014	Hanoi	054	Jungle	057
alib	002	Backup	169	Chess	096	CyclicSpace	255	DosKwik	129	FillDisk	061	HardCopy	075	Juggler	097
Alice	212	Backup	258	Chess	205	Cyberoids	049	DosPlus1	032	Find	134	Hash	020	Jumble	262
AInt	034	Badger	365	ChessTutor	269	CygnusEdDemo	095	DosPlus2	032	Find	197	Hawk	206	JustBeeps	239
AList	108	Balloons	272	ChinaChallenge	312	CZED	323	DoTil	070	FINDIT	168	HBBill	126	K1_Editor	332
AllocMaster	221	balls	001	Chop	302	D2D-Demo	082	Dos_Perfect	108	FINDIT	169	Hd	020	Kahnankas	116
AMC	182	Bally	181	CIATimer	178	dado	006	doty	001	FineArt	030	HeadDriver	098	Keep	149
AmCat	067	Bally	205	Circles	304	Dance	126	DPFFT	354	FirstSilCon	050	HeadClean	239	Keybit	274
AmBas	071	Bally	221	ClipPlane	101	Dark	087	DPFFT	324	Fix	015	Heart3D	236	Keyboard	005
AmICorm	157	Bally	232	Ciaz	087	DASM	168	DPhot	237	FiveInLine	163	Hed	164	Keyboard	291
AmICorm	178	Bankh	120	Clean	297	DASM	169	DPhot	290	Fix8010	241	HeliosMouse	094	Keyboard	344
AmICron	113	Bankn	163	Clear	105	Dasm	287	DPS	184	FixDisk	223	HeliosMouse	111	KeyFiler	157
AmICorm	139	banner	004	Cled	074	DataPlot	121	Dps	209	FixFd	183	FixHunk	001	KeyLock	081
AmICorm	082	BasicBoing	043	Cled	081	DataIoObj	172	gslide	011	FixHunk	197	Helper	332	KeyMac	253
amiga3d	012	BasicProgs	105	CLI-Colors	248	DateRequester	276	Draco	201	FixIcons	303	HexCalc	067	KeyMacro	354
amiga3d	014	BasicStrip	121	ClickDOS	276	Dazzle	015	Draco_Other	077	Fixobj	010	Hide	085	KeyMacro	325
AmigaBench	236	Batchman	325	ClickToFront	086	Dbug	002	Draco_SYS	076	Fixobj	015	Icon	130	KeyMapEd	388
amigademo	001	BattleForce	273	ClickFront	094	Dbug	041	DrawMap	229	FixObj	038	Ho-10c	038	KeyMapEd	193
AmigaDisplay	018	BattleForce	205	ClipIcon	167	Dbug	102	Drawmap	315	FixWB	059	HP11	153	Keymap_Test	019
AmigaFox	315	Bawk	065	CLimax	224	Dbug	232	DRES	168	FlamKey	105	HP11	274	KeyMenu	362
AmigaLine	138	Bawk	092	ClipBoard	056	DC10	150	DRES	169	Flex	156	HPMam	153	Kick	389
AmigaMonitor	040	BBChampion	244	CLIPrint	237	DCDemo	098	Drip	347	Flip	369	HR136	182	KickBench	234
AmigaMonitor	070	BBChampion	298	CL Utilities	162	DClock	388	Driver	039	Flip	390	IceFrac	303	KickFont	179
AmiGant	248	Bbm	043	Clock	015	DClock	298	DropCloth	059	Flip	248	Hide	085	KeyMacro	325
AmigaPunt	272	Bbs	030	Clock	036	DFrame	325	DropCloth	201	Flow2Troll	162	HyperBase	058	KeyMapEd	388
AmigaSpell	067	BbsList	043	Clock	059	DeepSky	219	DropShadow	059	Fm	036	HyperBase	131	Killer	115
AmigaTCP	225	BeachBirds	112	Clock	168	dhexes	010	DropShadow	074	FMC	244	HyperDialer	367	KillReq	290
amigalarm	001	beep	014	Clock	169	Dex	014	DropShadow	087	Fme	148	HyperNet	176	Kide	268
AmigaAtari	020	Berserk	100	ClockDJ	293	DEM	303	DropShadow	112	FME	346	Hypno	297	Klabor	332
AmigaTrek	278	Berserk	132	CloseMe	167	DEMolition	089	DropShadow	112	FMSdisk	294	Hypocydoids	027	KwikBackUp	234
AmigaTrek	320	Berserker	260	CloseWB	065	Demon	303	DrunkenMouse	105	Focus	112	IceFrac	303	L	105
AmigaVenture	041	Berserker	355	Cloud	216	Delab	179	DSM	167	FontEditor	030	Icon	081	Label	096
AmigaWave	211	Bezier	075	Clue	045	DevKit	297	DTC	335	FontEditor	073	Fontr	081	LabelPrint	277
AmiGazer	080	bgrep	004	CM	280	DevPatch	332	DTerm	073	Fontr	081	Font	081	LabelPrint	210
AmiGazer	090	BI	375	CManual	337	DevRen	378	Du	048	Foogd	066	Foreach	138	LabelPrint	238
AmiGo	289	BI	184	Cmd	079	Devstat	292	Du	345	Formatter	041	Formatte	377	LabelPrint	246
AmiOmega	320	BigView	058	Cmd	086	dex	014	DualPlayField	041	Formatte	377	Formulas	316	LabelPrint	363
Amoeba	120	Bind	309	Cmd	095	DeHexBin	321	DM2	075	Formulas	316	IconAssembler	101	LabelPrint3.5	363
AmSci	066	bison	004	CMF	168	Dif	131	DumpDiff	253	Fortune	369	IconExec	012	Labels	111
AMUC_Demo	117	Bison	051	CMP	169	Dframe	221	Dur5	067	Fortune	311	IconFrac	067	layers	005
AMXISP	181	Bison	105	CNewsBin	318	Dg210	040	DvorakKeymap	027	FontEditor	030	Icon	081	Label	096
AMyD	111	bm	004	CMNewsSrc	319	Dhystone	001	DWP	160	FontEditor	073	Font	081	LabelPrint	277
AnalogJoyStick	247	Bison	136	colorful	001	DICE	359	DX-Synth	082	FontEditor	073	Font	081	LabelPrint	210
AnalyticCalc	328	BlackBook	070	ColorReq	257	DIEd	388	DV-XoicesSorter	149	Fractal	052	IconImage	120	LabelPrint	238
AnalyticCalc	104	BlackBox	178	ColorReq	248	DiEdit	036	DynaShow	285	Fractals	371	IconImage	071	Launch	179
AnalyticCalc	144	BlackJack	019	ColorTools	323	DiEdit	179	EasyBackup	365	Frag	278	Icons	044	Lav	054
AnalyticCalc	176	Blank2	146	Colour	075	Diff	075	Easymouse	365	Fragit	107	Icons	067	layers	005
AnimalSounds	149	Blink	034	ColumnSet	270	Diff	087	Echo	036	FontEditor	073	Icon	071	id	010
Animations	053	Blink	040	Comal	296	Diff	107	echo	079	Frags	069	Icon2C	377	LeastSquare	105
AnimBalls	157	Bit	276	Comm	048	Diff	138	Echo	217	Free	388	Icon2C	124	LeatSquare	105
AnimPlayer	096	BitDemons	256	Comm	067	Diff	142	Echo	311	Free	066	Icon2C	151	LeatSquare	128
AniPtrs	332	BitLab	069	Comm	071	Diff	231	ECPM	157	Free	159	Icon2C	213	LeftyMouse	229
AniPtrs2	364	BitLab	084	Comm	075	Diff	281	Ed	084	freedraw	001	Icon2C	350	Lemacs	022
AniPtrs2	399	BitLab	191	CONFIG	168	DiGraph	073	EGraph	058	FTOHEX	168	Icon2C	316	Less	034
Append	379	Blob	358	CONFIG	169	DiMaster	088	EgyptianRun	120	FTOHEX	169	Icon2C	316	Less	074
Append	389	Blob	015	Comman	069	DiMaster	089	Elements	368	FullView	242	Icon2C	316	Less	074
AppData	301	Blocks	071	Comman	080	DiMaster	130	Elements	384	FullView	287	Icon2C	316	Less	074
Aquarium	040	Blocks2	156	Comman	081	DiMaster	130	Elements	384	FullView	287	Icon2C	316	Less	074
ArchEdge	362	bm	004	Comman	090	DirUtil	035	Emotions	027	FuncKey	088	Icon2C	31		

FishList
Alphabetical listing, by program name, of Fish disks 1 through 390
Compiled from 'Contents' files using Transcript, Professional Page, and AmigaCOMAL

Lotto	262	MiscUtils	302	OnePlane	329	PPShow	371	Rubik	029	SmallC	141	termcap	014	VirusUtils	331
LowMem	085	Missile	050	OnlyAmiga	126	PPShow	334	Rubik	285	SmallLib	092	Terrain	061	VirusX	337
LPatch	061	Mitb	183	OnlyAmiga	258	PPType	371	RunBack	065	SmallTalk	037	Terrain3d	094	VirusX	154
LPE	243	MM	327	OPPS	156	Pr	034	RunBack	152	SmartIcon	134	TES	162	VirusX	158
is	010	Modula-2	024	OPSSc	358	Prep	122	RunBack	214	SmartIcon	214	Tellic	324	VirusX	175
Ls	178	ModulaTools	094	Orbit	289	PrFont	055	RunBackGround	073	sMOVIE	316	Tellic	173	VirusX	216
Ls	236	moire	009	Othello	028	Print	184	Rxl	299	Smus3.6a	167	TeX	083	Virus Alert	154
LVR	331	Mon	310	Othello	080	printSupport	005	S220b8SVX	286	Smush	081	textdemo	005	VLabel	137
Mo2Amiga	113	Mon1DCMP	108	Othello	090	Print1.2	174	SafeBoot	175	SMusicPlayers	058	TextDisplay	244	Vit	202
Mo2Error	079	Monopoly	015	Overscan	133	printer	005	SafeMalloc	027	SNAG_Pointers	364	TextDisplay	188	Vit	226
M4	160	Monopoly	251	P2C	341	PrinterStealer	095	Sam	182	Snake	088	TextPaint	346	Vit	245
MacFont	138	MonProc	069	Painter	223	PrintText	080	Samp	307	Snake	089	TextPaint	361	Vit	257
Mach	136	MonProc	079	Pack	020	PrintText	090	SAMP	348	SnakePit	343	TextPlus	359	Vit	308
MachII	163	Moonbase	312	Pack-It	103	PrintHandler	282	samplefont	005	Snaps	088	TextPlus	375	VMK	328
MachII	254	MoreArt	106	PacketSupport	346	PrintHandler	352	SampleScanner	363	Snaps	274	Textra	239	VMore	085
MachIII	378	MoresBettar	224	ProcMan87	192	Printl	234	Sand	082	Soap	326	TFile	350	Vnews	085
Machine	109	MoreRows	054	PageCnt	304	PrintPop	108	SASTools	338	Snapshot	066	Thai	098	VNews	107
Mackie	131	Moria	194	PagePrint	040	PrintSpool	154	Sb	069	Snapshot	073	TheA64Package	379	VoiceFiler	038
Mackie	161	MoCalc	385	Paint	128	PrintStudio	366	SBProlog_1	140	SnipDemo	199	TheGuru	378	Vog	070
Mackie	187	Moska	330	PaintJet	129	PrivHndr	230	SBProlog_2	141	SnipIt	217	Tiles	241	VRTest	161
Mackie	189	Mounted	079	palette	001	Proc	236	Se	087	SnoodDOS	388	Tilt	054	Vsnap	326
Mackie	267	mounted	079	palette	055	ProgCalc	139	scales	006	Snooker	303	timer	005	Vsprites	055
Mackie	305	mouse	005	Palette	330	ProgCalc	172	ScanIFF	358	Sortfront	327	Timer	080	Vsprites	061
MacView	032	MouseBounce	204	PalleteReq	242	proctest	005	Scanner	218	Sortfront	342	Timer	090	Vt100	029
MacView	035	MouseClock	279	PatlTest	087	prof	009	Scat	081	SoftSpan	374	TimeRam	096	Vt100	036
Magnetic_Pages	372	MouseClock	069	Panl	082	ProfileMacros	046	SCAT	168	SoftSpan	343	TimeSet	071	Vt100	041
Mailch	324	MouseCords	287	parallel	005	Profiler	048	SCAT	169	Sol	103	Timetest	346	Vt100	047
make	002	MouseOff	073	Paranoids	250	ProgUls	302	SceneGenDemo	299	Solitaire	032	TinyProlog	145	Vt100	055
Make	045	MouseOff	075	ParM	375	ProgGadget	269	Scenery	155	SonicPeak	217	TitleGen	300	Vt100	114
make2	002	MouseReader	097	ParOut	073	ProSuite	107	Scheme	149	sortic	006	TitlePage	174	Vt100	138
Makelcon	255	MouseUtil	184	Parse	092	PrtDrivers	128	ScienceDemos	027	Sounddemos	167	Today	368	Vt100	275
Makemake	074	MovePointer	087	ParseNag	165	PrtDrvGen	060	SclSub	142	SoundEditor	355	ToolLibrary	376	Vt100	330
Makewords	366	MoveWindow	087	ParTask	079	PrtDrvGen	080	SCM	367	SoundExample	055	Tools	080	Vttest	035
MalloTest	066	Movies	116	PascalToC	065	PrtDrvGen	090	ScnSizer	079	Sobzon-C	171	TooMuch3D	270	Wanderer	216
Man	241	MRARPfile	281	Password	365	PrtSpool	216	ScotDevice	084	SpaceIceDemo	090	Touch	049	WarpText	087
Mandala	067	MRBackUp	327	Password	243	Ps	026	Scrambler	148	SpaceLog	305	trackdisk	001	WarpText	096
Mandala	387	MRBackUp	128	Paste	136	PshInp	101	ScreenSave	246	SpeakerSim	340	TrackDos	365	WarRanger	139
Mandel	004	MRBackUp	129	Patch	129	Pt_Anim	170	ScreenShare	308	Speed	237	Tree	075	WBDualPF	087
Mandel	111	MRBackUp	170	Patch	296	Purity	066	ScreenShift	088	Speech_demo	005	Tracker	181	WaveBench	283
Mandel	218	MRBackUp	270	PatchLoadSeg	331	PusOver	122	ScreenShift	089	SpeechTerm	020	TrackSave	312	WaveMaker	318
MandelBlitz	387	MRBackUp	279	PatchNTSC	346	PUZZ	362	ScreenX	158	specchtay	001	TrackSave	355	WBAssign	277
Mandelbrot	005	MRMan	281	PatEdit	130	Puzzle	032	ScreenZap	303	specchtay	005	TrackUtils	350	WBColors	121
MandelBrot	021	MRPrin	167	Path	177	PuzzlePro	122	ScreenZap	157	specchtay	203	Trails	032	WBD	346
Mandelbrot	031	MRPrin	279	PathMaster	245	PWDemo	071	Scrimper	018	Speed	237	Tree	031	WBDepth	175
Mandelbrot	239	MSDOS	158	PatternLib	254	Pyro	199	Script	288	SpeedDir	075	Tree	306	WBShadow	253
MandelBrots	020	Msh	382	PCBTool	158	Pyth	238	ScrollPt	035	Spew	069	Trees	049	WBDump	058
MandelMountains	295	Msh	327	Poopy	383	Pz15	191	SCT	151	Spiff	172	TreeWalk	352	WBlander	100
MandelMountains	354	MSizer	285	Poopy	151	QBase	098	SDBackUp	128	Spigot	272	TreeWalk	289	WBlander	114
MandelMountains	383	MuchMore	378	Poopy	243	QMan	130	SeaHaven	260	Spin3	032	Trek73	010	WPic	234
MandelVroom	078	MuchMore	234	PcPatch	163	QMouse	049	Sectorama	102	SpinPointer	269	TrekTrivia	180	WBRes	277
MandelVroom	214	MuchMore	253	PcPatch	218	QMouse	262	Sectorama	108	Spines	057	TrekTrivia	252	WBRun	043
MandelVroom	215	MuchMorePoPa	378	PCQ	183	Q2	186	Sed	128	Spines	097	Tricky	367	WBShadow	253
MarbleSlide	283	mult	006	PCQ	339	Quattro	230	Send	231	Spool	069	TriClops	035	Wc	069
Marge	177	Multic	304	PcView	164	QueryAny	079	SeeHear	335	SpreadSheet	053	Triple	206	Who	179
Mariateroid	115	Multicalc	166	PDC	351	QuickCopy	035	Selector	302	SpriteClock	043	Tran	355	WFrags	131
Mat	374	Multid	049	Pdc	110	QuickFix	106	SendMorse	175	SpriteEd	040	Trunc	179	Whats	334
Match_Stuff	102	Multidim	018	ptherm	014	QuickHelp	302	SendPackets	035	SpriteEd	035	Tsize	035	Wheel	067
MathTrans	319	Multipl	292	PearlFont	061	QView	262	SendPackets	066	SPUDclock	165	TShip	179	WheelChairSim	139
Matlab	267	Multip	333	Pemacs	022	RadBoogie	269	SensorPro	283	Spy	369	TUC	150	Whereis	321
Matrix	376	Multip	373	PennyWise	273	RainBench	080	Sentinel	263	SentUp	010	Tunnel	174	Whezel	321
Maze	171	MultiSelect	253	PenFallont	073	Rainbow	090	SendUp	264	Session	051	TunnelVision	036	Wic	079
Maze	218	MultiTasking	020	Pere-Fits	105	Rainbow	058	Set	005	Slack	105	Tron	355	Wif	224
Maze	248	MunchingSq	087	PerfectSound	050	RamCopy	165	SetAlternate	012	StarBlanker	308	TurboBackup	139	why	079
MC68010	018	Munchie	137	Perl	161	RamSpeed	031	Setback	107	StarCharl	159	TurboMandel	302	Wicon	154
mcCAD	056	Mv	219	PersMail	067	Random	020	SetClock	390	Stars	118	Turnite	361	Wild	043
mcCAD	059	MVP-FORTH	009	PetersQuest	224	RandSam	074	SetCPU	187	StarTerm	012	Turtle	321	WireDemo	118
mcCAD	074	MWB	065	PetCLI	071	Random	074	SetCPU	223	StarTerm	030	Txed	031	With	253
Mean18	304	Mx.Example	031	Pex	206	RawIO	085	StarTerm	101	StarUps	101	TxitExe	286	wKeys	128
Med	060	McGade	052	PickPacket	227	RayTracePics	044	StFont	041	Steve	197	Update	045	X-Spell	040
MED	255	MXMlib	354	PictureGarden	206	RayTracer	066	StFont	075	Steve	217	Update	254	X2k	345
MED	349	MyLib	046	pictures	011	RCS	282	SetKey	070	Steve	256	Update	286	Xboot	161
MegaWB	253	MyCLi	004	Pictures	045	ReadmeMaster	163	SetWindow	012	Steveslag	238	Update	365	Xcolor	184
Merf	066	MyMenu	225	Pictures	072	ReadmeMaster	390	Sh	174	Stitchery	350	Ueturbo	311	XCOLOR	361
MemBoardTest	158	MyUpdate	049	PlayList	018	Record-Replay	105	ShanghaiDemo	055	StockBroker	343	UnHunk	026	XCopy	074
MemClear	058	Nag	161	Plot	035	RemLib	139	Shar	092	STRReplay	258	Unlidel	035	Xebec	224
MemDrag	214	NameGame	256	PipeDevice	055	ResourcerDemo	232	ShortCut	090	StringUb	029	Xenozip	300	X-Hair	234
MemEuler	366	Names	122	Piperander	084	ResourceDemo	232	ShortCut	229	SupLib	258	Xlens	074	Xicon	031
MemExpansion	027	Nart	066	Pipeline	165	Relab	389	SID	338	Surf	170	Xlips	092	Xlips	073
MemFlick	206	NComm	356	Regexp	222	relob	107	Show	323	Surf	170	Xlips	109	Xlips	073
MemFrag	345	NComm	230	Regexp	311	reqBut	332	Supermart	321	Surveyor	151	Xlips	184	Xlips	073
MemGauge	222	NComm	246	Region	121	Repring	180	SuperMart	321	ValBench	184	Xlips	046	Xlips	073
MemGuard	354	NDebt	366	Requester	175	RevBut	332	SuperReLab	304	VCheck	126	Xlips	247	Xlips	073
MemGuard	325	NDir	092	Repliers	084	Reversi	038	SuperReView	367	VClock	126	Xlips	310	Xlips	171
MemLib	240	Nemesia	127	PopCLI	040	Reversi	305	SupLib	126	VDraw	313	Xlips	228	Xlips	003
MemRoutines	362	NeuronNets	326	PopDir	204	Robotoff	059	SupLIB	168	VDraw	313	Xlips	038	Xlips	039
Mem Trace	163	NewDemos	080	PopDir	111	Ripples	127	SimCPM	109	VAXIerm	336	Xlips	053	Xlips	018
Mem Watch	048	NewEx	368	PopDir	204	Robotoff	082	SimGen	243	SiylInfo	364	Xlips	092	Xlips	073
Mem Watch	087	NewFonts	034	PopDir	287	Rocket	085	SignFont	211	Surf	145	Xlips	345	Xlips	135
Menu	387	NewFonts	071	PopDir	172	RoadRoute	328	Si	102	Surveyor	151	Xlips	079	Xlips	073
MenuBuilder	065	NewFonts	071	PopDir	204	RoadRoute	358	SizeChecker	244	TACL</					