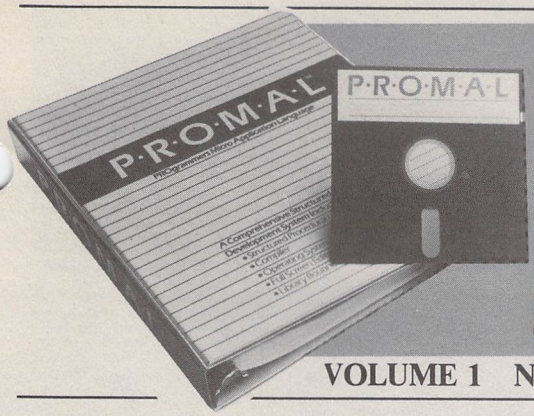


PROMAL™ NEWS

PROgrammer's Micro Application Language

VOLUME 1 NUMBER 2 July-September, 1986

PROMAL 2.1 WITH SOURCE CODE AVAILABLE!

You read it right! Now you can get PROMAL with Source Code!

By the time you read this, a new, improved version 2.1 of PROMAL should be available for the Apple IIe, IIc and Commodore 64. But the most exciting news about PROMAL 2.1 is that, for the first time, you will be able to order the complete SOURCE CODE for the EXECUTIVE, EDITOR, and GRAPHICS TOOLBOX, as well as a complete source listing of the Runtime package and library!

With the source code, you can explore the inner workings of the EDITor and EXECUTIVE and customize them if you wish. Since both these programs are written in PROMAL, you can change them with the EDITor and re-compile them to personalize your program development system!

Maybe you'd like to change the EXECUTIVE's prompt (-->) or choose different control keys in the EDITor. With the source code, you can!

There's more good news. Even if you don't get the source code, you'll want to take advantage of our low-cost upgrade offer for owners of earlier PROMAL versions. Here's a rundown of some of the new version 2.1 improvements over version 2.0:

Apple II PROMAL 2.1

- Full support for /RAM disk. Now you can use all that memory (128K to multi-megabytes!) in your IIe!
- PREFIX command now finds volume name for any slot/drive.
- Full support for all path names (including '.' in volume name and single character file names).
- Refer to disks by drive or prefix (for example, COPY 2: MYFILE.S 1: copies from drive 2 to 1 without specifying the volume name). 0: is /RAM disk!
- New GETKEY function gets key with cursor but without echo to screen.
- Larger workspace.
- Improved EXECUTIVE, LOADer.
- More free memory space.
- All known bugs fixed.
- All-new improved manual (not just page changes).
- Source code available (optional).
- Much more!

Commodore 64 PROMAL 2.1

- DYNODISK revamped for super-reliability.
- New GETKEY function gets key with cursor but without echo to screen.
- Improved EXECUTIVE, LOADer.
- More free memory space.
- All-new improved manual (not just page changes).
- Source code available (optional).
- Much more!

/continued page 3

IBM PROMAL NOW AVAILABLE— WITH SOURCE CODE

The long-awaited PROMAL system for the IBM PC family of computers and true compatibles is now available. Complete source code for all elements of the system, including the Compiler, Editor, and Runtime Package, is also available

The PROMAL Compiler for the IBM family runs under DOS 2.0 or higher and produces very fast native 808X machine language output. Most programs written in Apple or Commodore PROMAL are readily portable to the IBM PC, and software is provided to transfer files between computers using the serial port.

In addition to the normal PROMAL features, the IBM version has a number of enhancements including:

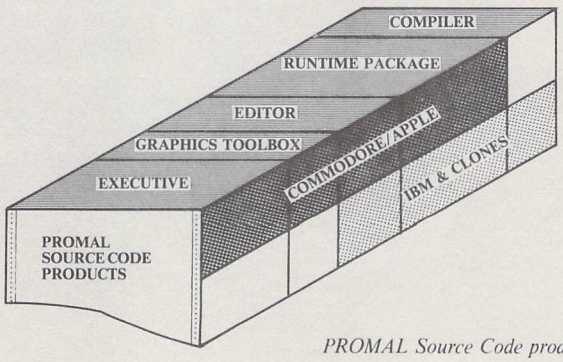
* Direct DOS calls
* Direct ROM BIOS calls
* Access to all 640K of memory
* Direct I-O port access
* DOS command line and Environment access
* In-line machine language code can be inserted in PROMAL source code.

PROMAL programs on the IBM execute faster than Turbo Pascal 3.0. The Compiler is unbelievably fast, typically processing 4500 lines a minute on an AT.

The Editor is very similar to the Apple/Commodore version, but scrolls to display lines of over 80 characters. The Editor generates standard DOS text files.

The main difference between Apple/ Commodore PROMAL and IBM compatible PROMAL is the absence of the Executive. MS-DOS provides the functions of the Executive in the IBM environment. If you are not familiar with MS-DOS, don't worry; the commands are very similar to the EXECUTIVE.

/continued page 3



PROMAL Source Code products currently available.

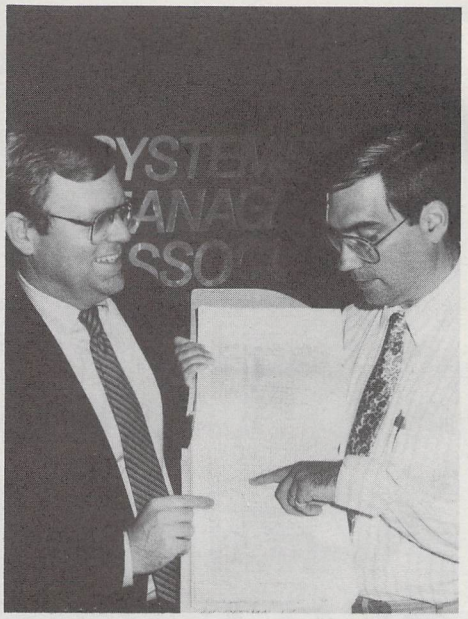
The Last Word

Dear Friends:

Are we crazy or what to be selling source code? Anytime you do something no one else has done you do feel a little strange. Frankly, we decided that if multitudes of you programmers are going to convert to using PROMAL you need more "goodies" than just the elegance and performance of PROMAL alone. After all, you do have to learn a new language, and that takes real commitment.

So we decided to include the best of all possible "goodies" in our PROMAL product line—source code. Now you can see hundreds of examples of PROMAL coding techniques by reading the source. That should make mastering the language even easier. Plus, you can customize the Executive or Editor to suit your personal needs perfectly. You can even use Runtime source package to make enhancements or extentions to the system.

You can now have total control of your programming environment — because your friends at SMA are just a little crazy. □



Jack Segner, President, SMA, Inc. and Bruce Carbrey show off new PROMAL Source Code products.

PROGRAMMING TIPS & TECHNIQUES

by Bruce Carbrey, SMA, Inc.

COMPUTING GREATEST INTEGER

There are a number of applications where you need to compute the greatest integer of a real number, such as rounding or finding only the decimal part of a number. This function is called INT in Pascal and many BASIC dialects; C programmers know it as FLOOR(). For positive arguments less than 65536.0, FLOOR(X) can be simulated in PROMAL using

```
(X:+):.
```

which casts X to an integer and back to a real. For values outside this range, however, we need a more general function. For example:

```
FLOOR(-3.8)          ; should return -4.0
FLOOR(100000.89)     ; should return 100000.0
```

The FLOOR function is included in version 2.1 of PROMAL, but owners of earlier versions can use the code shown at right.

Space does not permit a detailed explanation of the workings of FLOOR, but the concept is to compute the binary exponent from the internal floating point representation and zero all bits to the right of the implied binary point in the mantissa. □

```
FUNC REAL FLOOR ; (Val)
  ; Return (real) largest integer less than or equal
  ; to the (real) argument.
ARG REAL VAL   ; Argument
REAL NEWVAL    ; Result
WORD I         ; Loop counter
BYTE EXP       ; 8 bit exponent
OWN BYTE TEMP[6]
DATA BYTE MASK[]=$FF,$FE,$FC,$F8,$C0,$E0,$C0,$80
BEGIN
IF VAL < 1.0 AND VAL > -1.0
  IF VAL >= 0.
    RETURN 0.
  RETURN -1.
BLKMOV #VAL,TEMP,6
EXP=TEMP@< (<1 + (TEMP[1] >= $80) -$7F ; exp.
IF EXP >= 39
  RETURN VAL ; Keep all bits of mantissa
EXP=39-EXP   ; # Bits of mantissa to clear
I=5
WHILE EXP >= 8 ;Can clear whole byte?
  TEMP[I]=0
  EXP=EXP-8
  I = I-1
TEMP[I]=MASK[EXP:+] AND TEMP[I] ;Clear bits
BLKMOV TEMP,#NEWVAL,6
IF VAL < 0.
  IF NEWVAL <> VAL ;Not already an integer?
    NEWVAL = NEWVAL - 1.
RETURN NEWVAL
END
```

IBM */from page 1*

Source Code Available

Source code is available as an option, allowing you to explore the inner workings of PROMAL or to "customize" the system to your personal taste. The EDITOR and COMPILER are written in PROMAL (of course), so you can customize them if you wish by simply editing and re-compiling. The Runtime Package source consists of about 9,000 lines of 808X assembly language which can be assembled with the Microsoft assembler (MASM).

An all-new 250 page manual is included along with numerous demo programs including a simple data base management program, interrupt-driven communications programs, and a windowing demo. The windowing program includes routines you can use to generate menus, control video attributes (color, reverse video, etc.) and create professional looking screen displays.

The current release of the IBM PROMAL compiler is Version 1.9, which directly generates executable programs (.EXE files) of up to 64K of program plus 64K of data. Purchasers will receive a free upgrade to version 2.1 in the Fall, which supports unlimited code size, separately compiled modules and is compatible with the Microsoft Linker and symbolic debuggers.

If you write scientific or engineering applications, you'll be pleased to know that the 8087 floating point coprocessor is fully supported in version 2.1 for fast, double-precision arithmetic. Standard PROMAL 6-byte floating point arithmetic is also included. A new 32 bit integer data type has been added, too.

Version 2.1 also supports the optional Graphics Toolbox for CGA, EGA, and compatible video adapters. The graphics toolbox makes it a snap to produce fast full color graphics displays. Routines are also provided for controlling a mouse (Microsoft Mouse, Mouse Systems, etc.).

PROMAL is designed to run on the IBM PC family of computers (PC, XT, Jr, AT etc.) or true compatibles (Compaq, AT&T 6300, Tandy 1000, etc.), with at least one floppy disk and 192K of memory. IBM compatible PROMAL is available only in the Developer's version, which means you can generate stand-alone programs and distribute them as you see fit without any royalty payments.

Please see the order form for complete pricing information. □

SOURCE CODE: WHAT'S IN IT FOR YOU?

As you can see, we're pretty excited about being able to make the source code for the PROMAL system available to you. But it may not be obvious to you what the benefits are of having the source code available. Also, you may not be sure which source code you need, if any. Let's try to answer these questions.

There are lots of things you can do with the source code, depending on your needs, but here are a few.

First of all, the source code to the EDITor and EXECUTIVE provide a wealth of useful programming examples written in PROMAL. There's no better way to improve PROMAL programming than by studying some real programs. For example, by examining the EXECUTIVE you can learn how the COPY command works or how command line arguments are processed.

Secondly, you can make minor changes to the system to suit your own taste. For example, suppose you don't care for the EXECUTIVE's "`-->`" prompt, and would like to change it to say "EXEC:". All you would have to do is EDIT EXEC.S and use the FIND command to locate "`-->`", which you would find defined as follows:

```
DATA WORD SYSPROMPT = "\0D--> " ; Command prompt
```

Just edit this line to read

```
DATA WORD SYSPROMPT = "\0DEXEC: " ; Command prompt
```

and save the file. Then just compile EXECUTIVE, and the next time your boot your system, you will be greeted with your custom prompt!

Of course not all changes are this simple, and there are limits to how much you can add without using too much memory, but there are lots of things you can do to customize the EXECUTIVE or EDITOR.

What About the Runtime Listing?

What is the Runtime Package/Library Listing, and why would you want it?

First of all, to make effective use of the Runtime source listing, you'll need at least some familiarity with assembly language programming, because the runtime package consists of over 10,000 lines of assembly language statements. It contains all the library functions and procedures (PUT, GETL, GETBLKF, LOAD, OUTPUT, etc.), plus the floating point arithmetic routines, error handling (runtime errors, ABORT, etc.) and, at the core, the instructions which actually execute your program.

Armed with the Runtime listing, you can create simple patches to the system to change the way things work. For example, suppose you wanted to have some key other than CTRL-B perform the prior-line-recall function. By examining the listing of the GETL, EDLINE, or INLINE routines, you will see that you need to change the value of a variable called BKEYBT which is defined on line 1146 (for example) of the listing like this:

```
1146 1010 02   BKEYBT .BYTE $02   ;Ctrl/B=backtrack
```

This tells you that this variable is at address $1010 and its value is $02 (the ASCII code for CTRL-B). To change it to say, CTRL-R, you could simply patch address $1010 to $12. You could do this right from the EXECUTIVE or in a JOB file with:

```
SET 1010 12
```

or from within a program with:

```
EXT BYTE BKEYBT AT $1010
...
BKEYBT = $12
```

Experienced programmers will be able to make sophisticated changes in the way the system works with the aid of the Runtime listing.

So What Do You Need to Buy?

If you're a commercial software developer, you'll definitely want to get the Developer's "Package Deal" (see order form). If you're familiar with assembly language, but don't plan on selling stand-alone software, then you'll probably want to get the End User's package deal. If you don't understand assembly language and don't intend to learn, then you might just want to get the source for the individual EDITOR and/or EXECUTIVE for your system. Either way, we think you'll find the PROMAL source code an invaluable asset in your collection of programming tools.

PROMAL 2.1 */from page 1*

About The Source Code

For the first time, we are offering the complete source code for the PROMAL 2.1 EXECUTIVE and EDITOR, as well as a listing of the Runtime package.

The source code is supplied on disk as PROMAL programs you can EDIT and COMPILE. For advanced programmers, a complete, commented, assembled source listing of the runtime package and library is available. This listing has about 800K bytes on six or seven disks (depending on your system), including an alphabetized cross reference map. You can print or display all or parts of the listing for a ready reference.

The source code for the SGD (screen graphics drivers) section of the Graphics Toolbox is also available as a 6502 assembly language file and assembled listing on disk.

Each source code package includes a brief Theory of Operation manual on disk as well as the source files.

Please see the order blank for pricing information for PROMAL 2.1 and the optional source code.

We are also considering making a cross-development package available which would run on the IBM PC and allow PROMAL compilation for the Apple/Commodore (including the ability to compile the compilers and assemble the 6502 runtime/library). If you are interested, please call and let Peter or Bruce know. □

Don't Forget The Public Domain Disks!

The first two Public Domain Disks for Apple and Commodore are still available. We have received several donations for our next Disk and we'll have a new one out very soon. ATTENTION APPLE DEVELOPERS: We're a little short on public domain contributions for the Apple, so get yours to us ASAP!

These are contributions by users. You can freely use or copy them, SUBJECT ONLY TO ANY RESTRICTIONS IMPOSED BY THE PROGRAM AUTHOR.

SMA serves as a clearing house for the PPDL (PROMAL Public Domain Library). We do not endorse or support the programs. Don't even think of calling or writing us about PPDL problems.

All programs are written in PROMAL and supplied in both source and compiled form. Most have documentation files supplied by the author. A few have no documentation, but are easily understood by a look at the source code. Hardcopy documentation is provided where necessary.

Disk #1 for the Commodore:

Contents:

1. Macro assembler, by C. Martens
2. Disassembler, by Steve Vermeulen
3. Disk Fixer program, by A. Ryan
   Includes a twenty-page manual describing Commodore disk structures and how to change them using the disk fixer.

The delivered product consists of a disk and summary of contents, plus the Disk Fixer documentation in hardcopy form.

Disk #2 for the Commodore:

Contents:

1. Document Formatter Program, by David Long
   A powerful and well established method of word processing using a maually created file containing text and formatting commands for such things as margins and page size, paragraphs and "absolute" lines of text, underlining, page headers and footers and page numbering. The file is fed into the document formatter program which produces a printer-ready result. A file is included which, when fed through the document formatter, produces the 15-page user's manual which is also supplied in hardcopy form.
2. Screen creator, by Rev. Mike Cargill
3. Printer control issuer, by Julia Christianson
4. PROMAL source file lister, by Garth Ingram
5. Screen creator, by W.A. Marsh
6. Graphics routines and demo, by Roger Norrod
7. Lister that includes time and date stamp, by Michael T. Veach
8. C64-to-Tandy PC2 data exchange program, by Steve Vermeulen
9. "Dumb terminal" emulation routine, and demo, by Steve Vermeulen
10. KOALA touchpad support, by Erik Vigmostad
11. Counter of word occurrences in a file, by Erik Vigmostad
12. File lister for RS-232 printer, by Erik Vigmostad

The delivered product consists of a disk and summary of contents, plus the Document Formatter documentation.

Disk #1 for the Apple:

Contents:

1. Macro assembler, by C. Martens
2. Disassembler, by Steve Vermeulen
3. Printer control issuer, by Julia Christianson
4. PROMAL source file lister, by Garth Ingram
5. Counter of word occurrences in a file, by Erik Vigmostad

The delivered product consists of a disk and summary of contents.

Disk #2 for the Apple:

Contents:

1. Document formatter program, by David Long. It is described above under "Disk #2 for the Commodore", item 1.

The delivered product consists of a disk and summary of contents, plus the documentation in hardcopy form. □

SYSTEMS MANAGEMENT ASSOCIATES
3325 Executive Drive, P.O. Box 20025
Raleigh, North Carolina 27619
Telephone: (919) 878-3600