HOW TO GET THE MOST OUT OF GEOS™

Jim Oldfield, Jr.
and
Tim Sickbert

Midnite Press/Midnite Software Gazette - the PAPER
Micro-PACE Computers, Inc.
Robert Wolter, Publisher
James A. Oldfield, Jr., Editor-in-Chief
Timothy B. Sickbert, Editor

Champaign, Illinois, USA

Cover Art: THE WORLD, sculpture from the C. Jere' collection, Artisan House, Inc., Los Angeles, California. Photographed by Wilmer Zehr, Champaign, Illinois.

PREFACE

GEOS (tm) has caused quite a stir since its introduction by Berkeley Softworks and Commodore Business Machines. The advent of a sophisticated operating system coupled with ease of use may place GEOS in the all-time best seller list of the CBM 64 and 64C. Of course, being packaged with every 64C from the factory helps.

So it was that we felt the need to expand on the GEOS User's Guide, and "How to Get the Most Out of GEOS" was born - but oh how little we knew of this new operating system. Dissecting, analyzing, probing, delighting, brow beating into the wee hours; you have the opportunity to read what we have discovered. And, hopefully, you also will expand on this.

GEOS represents a quantum leap for new computer owners. An effective and efficient interface, and with CBM behind it, GEOS will be around for a while.

The Midnite Software Gazette/the PAPER are the oldest magazine in North America for Commodore brand computers. We trace our roots all the way back to 1979.

We at the Midnite Press/Midnite Software Gazette are, unlike many personal computer publishers, users of Commodore products just like yourself. We use nearly every Commodore machine to write and publish our books and magazine. Using PETs to VICs, C64s, C128s, Amigas, and PC10s, we prove the versatility, power, and economy of owning and using Commodore computers for both business and pleasure. Please enjoy the book.

James A. Oldfield, Jr.

INTRODUCTION

GEOS is a welcome addition to the Commodore software market. It holds attractions for the novice user, the advanced user, and the programmer alike. As a new operating system, GEOS needs support from all quarters. And users from all quarters need to do more than GEOS alone provides for.

"How to Get the Most Out of GEOS" provides support for the users and programmers who want to extend the usefulness of GEOS. Depending on where you start, you may need only a more comprehensive explanation of GEOS, utilities to make GEOS more open and portable, or you may need enough information about the inner workings of GEOS to write your own utilities and programs.

This book is not everything you might hope for. No book ever is, and no book ever will be. The best teacher is your own effort. We cannot supply you with the answer to every question, nor detail every routine of the GEOS code. We can give, and have given, a comprehensive overview that will serve you well as a starting point and a reference. Here are some tools that will serve you to extend your productivity and that you can use as an example for developing your own utilities.

As you discover and develop, share. It doesn't matter too much whether you go through the public domain or commercial channels, as long as the information gets out. If Berkeley and Commodore are going to bill GEOS as an operating system, the users and the programmer's need to treat it as an operating system. Sure, Berkeley, Commodore, and Midnite all hope to make some money off it, but the money should come from serving the users well.

Enjoy GEOS, enjoy the book, and share!

Timothy B. Sickbert, Editor

TABLE OF CONTENTS

WORKING WITH GEOS

SECTION 2

SECTION 3

Appendixes

How to Get the Most Out of GEOS

First Session

Welcome to the world of GEOS! Berkeley Softworks has, with the introduction of this new package for the Commodore 64, provided a new operating system that is more intuitive and in many ways more flexible than standard Commodore BASIC. The GEOS user interface, deskTop, presents information regarding the status of the system and disks quickly and clearly. The operating system uses the disk drive as a virtual storage device that allows you to access and manipulate more data than would fit into the computer at any one time. Such are the strengths of GEOS.

GEOS is also rather new. The operating system is neither fully understood nor fully documented and so may cause some confusion among users and programmers. The application programs and the method in which these programs store their data are a standard unto themselves. This makes it more difficult to import data into, and to export data from, the C64's new operating system. Such are the weaknesses of GEOS.

We shall, in this book, supplement and expand on the information given by Berkeley Softworks regarding GEOS. The information contained within this book has been gained primarily from using and digging into GEOS. A few pieces of information, or leads to find the information, have been gathered from the public domain - we have listened to the comments that many users have made either to us directly or to users in general via bulletin boards and telecommunication services.

This book includes descriptions of proprietary information. These descriptions are given in order to enhance the usefulness of GEOS. The Midnite Press/Midnite Software Gazette, the authors, and the contributors neither encourage nor condone the use of the descriptions contained herein for any illegal, unethical, or immoral purpose. In other words, do not steal. Berkeley Softworks has invested, and continues to invest, a great deal of money to provide an alternative to Commodore 64 users. There is no justification for using GEOS or any other commercial program without paying for it. We ask that if you use GEOS, buy it; if you use any of the supplements such as FontPak or DeskPak, please buy these as well.

WORDS

Even before you turn on your computer, you need to be familiar with just a few words that we will be using throughout the book.

input device: a joystick, mouse, trackball, lightpen, or sketchpad. GEOS V1.2 directly supports only the joystick as an input device, but the Commodore 1350 Mouse and various trackballs can be used as a joystick. At the time of this writing, Berkeley Softworks has released drivers for the Inkwell Flexidraw Lightpen (tm) and for the Koala Technologies Koala Pad (tm). The input device must be plugged into the controller port on the right side of the computer, nearest to you as you face the front of the keyboard.

fire button: this button on an input device sends a special signal to the computer. On a joystick, the fire button is usually on the end of the stick, on the base, or both. On a mouse, the fire button is usually at the front (back?) of the mouse, toward the tail. If a mouse has two or more buttons, try the leftmost button first.

pointer: the blue arrow that appears within GEOS. You can move the pointer using a joystick, trackball, or mouse.

click: to press the fire button on the joystick, trackball, or mouse.

double click: to quickly press the fire button twice.

icon: a graphic that represents a disk, file, or a function.

window: 1) a box on the computer monitor screen that is functionally separate from the screen that surrounds the box. In this case, a what goes on the window usually does not directly affect what is going on behind or around the window. 2) the use of the computer monitor screen to show a part of a document file that will not fit all at once. In this case, it is often referred to a window on a virtual page, and the window can usually be moved against the background of the virtual page. Think of holding a magnifying glass up to the small print on your insurance policy. The magnifying glass is the "window," and the policy is the virtual page.

toggle: this adjective describes features that are either on or off. For example, bold text can be toggled on in geoWrite by selecting Style/bold, and later toggled off by selecting the same command.

^ (caret): this character will sometimes be shown when talking about entering certain characters. It means that to enter that character you are to hold down the <CONTROL> key at the left side of the keyboard and then press and release the key that follows the ^

*: the asterisk will occasionally be used as a wildcard to signify that any of a range of commands may be used.

( ): e.g. (font) this notation will be used in place of a the asterisk wildcard, when the use of the asterisk would make the reference ambiguous.

select: to place the pointer on an icon or other feature on the screen, and click or double click, depending on the context and your intent. When selecting a command, you may need to move the pointer to the primary command, click once, and then move the pointer to the secondary command and click again.

drag: to place the pointer on an icon, click once, wait a moment, and then click again. This will cause GEOS to attach a shadow of the icon to your pointer, allowing you to "drag" the icon to the border, the printer, or the Waste Basket.

FIRST SESSION

If this is your first time using GEOS, please do two things before you start working with GEOS itself. Make at least one backup and at least one work disk. GEOS is a disk based operating system and as such is relatively easy to lose. Making a backup copy of the original GEOS disk is some insurance against future problems. Further, each of the disks that you use with GEOS will need to have some GEOS files on it and will need to have some special data stored on the disk. It is not a good idea to use the original GEOS disk or an only backup as a work disk.

MAKING A BACKUP

When you first take the GEOS disk out of the box, please put write protect tabs over the one or two little notches on the sides of the disk. These two notches allow the disk drive to write information to the disk, and so are known as write enable notches. A write protect tab is a small piece of opaque tape, usually metallic silver, sometimes black, that is included with a box of new disks. Covering the notches in the sides of the disk tells the disk drive that it is not to write to the disk. The original GEOS disk does not have room for any more information and it has several files that you will need sooner or later, so it is best to protect it from the beginning.

You may use any disk backup program to make a backup of the original GEOS disk. All that matters is that the backup program copy ALL the information on the disk. GEOS uses a non-standard file structure, so a file copy program will not transfer all the information you will need. A utility that copies all the blocks allocated in the BAM (block allocation map) will work. Good backup programs include Mr. Nibble, Fast Hack'em, DiSector, and SuperKit 1541. If you do not have such a program, use the Backup program provided on the original GEOS disk, or load GEOS and select the Disk/copy option from the Command Menu in the upper left corner of the screen. Both the Backup program and the Disk/copy option are relatively slow, and it would be well worth your while to find a good public domain copy program or to buy a commercial copy program.

Make at least two copies of the original GEOS disk. Later, you can make several work disks tailored to your own needs, but it is necessary to make at least two copies before you do anything else. One of these copies will be a backup, and the other will be a work disk. It is generally a good idea to store at least one backup copy somewhere away from the computer - like at the other end of the house. If anything should ever happen to the original GEOS disk, the backup copy can be used to restore the original disk or to make a disk that will act just like it.

If you are having a problem making a backup of the original GEOS disk, please refer to the appendix on Erros & Problems.

WORK DISKS

A work disk is the same as a backup disk except that you are free to do more with it. The purpose of a backup disk is to have copies of all the original GEOS files in case anything should ever happen to the original GEOS disk. With a work disk, however, it is important to have only the files that you will need for the work that you wish to accomplish.

PLAYING AROUND

At this point, with write protect tabs on the original disk, the backup stored away, and at least one work disk in hand, you cannot do any permanent damage short of feeding the original GEOS disk to your dog or taking a hammer to your computer system. You can leave all fear behind because any mistake you make at this point can be recovered. You are free to play around.

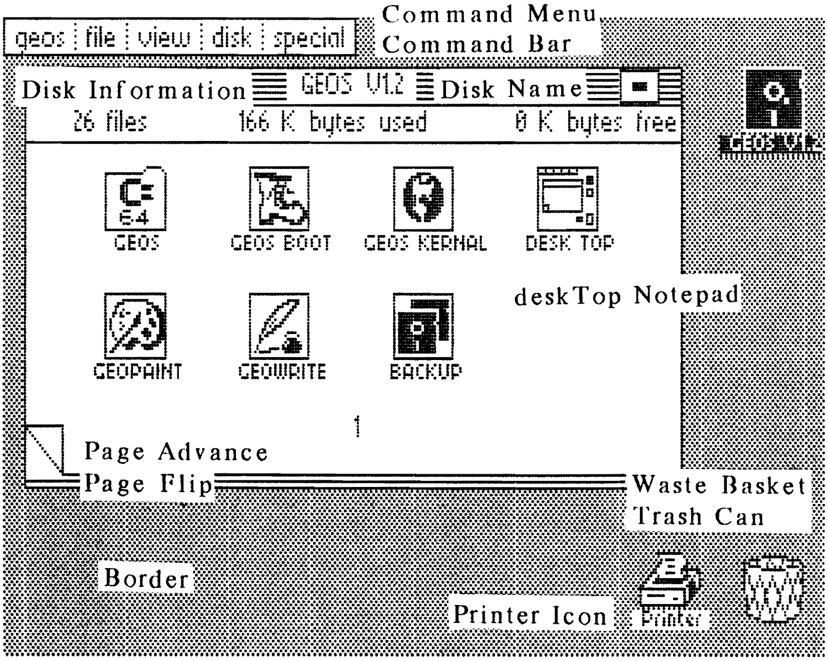
This particular work disk is quite important - it is yours to do with as you please. You have write protect tabs on the original GEOS disk and a backup stored away. Use this work disk to try things, to drag icons around, to flip through the pages of the deskTop notepad to see what all is on the disk, and to try the options in the command bar. This work disk is important because you can do anything you want to with it without being afraid of breaking anything. And the only way to learn how to use GEOS is to play with it.

GEOS has much to offer, and finding what you need is usually a matter of using your intuition to stretch the obvious. But taking the time to look around will give you a better feel of where everything is at.

If you have not already loaded GEOS, do so now. Turn on your monitor, printer, disk drive, and computer. Place the original GEOS boot disk (be sure that you have the write protect tabs in place) in the disk drive, and either:

```
    LOAD "GEOS",8,1 <RETURN>
or
    LOAD "GEOS",8 <RETURN>
    (wait for the flashing cursor to come back)
    RUN <RETURN>
```

In a few moments, GEOS will present you with the deskTop screen. We shall follow the conventions for referring to the features of GEOS established in the GEOS User's Guide except where the User's Guide is incomplete, inaccurate, or ambiguous as noted in the illustration below.



For example, in the upper left of the deskTop screen is the Command Menu. Selecting one of the five options in the Command Menu will cause a sub-menu to drop down. These could be referred to as primary and secondary commands, or any of a number of other terminologies could be used. In this book, we shall refer to a command by giving the sequence to be selected in a shorthand form. That is, to exit GEOS and go back to Commodore BASIC, you would select Special/BASIC. This means that you are to move the pointer to the box on the Command Menu that contains the word Special, click the fire button on your joystick or mouse once, move the pointer to the box in the sub-menu that contains the word BASIC, and click again. The shorthand method saves me some typing.

Before you play around too much, you need to exchange your work disk with the original GEOS disk. For all the magic that computers seem to be able to do, you need to tell GEOS that you are swapping disks. To do this, move the pointer to the little box that is in the upper right corner of the deskTop notepad. Click once on this box and the deskTop notepad will clear and the disk icon to the right will change from a disk to a question mark. Now insert your work disk, move the pointer to the disk icon, and click.

Take some time now to explore the features of the deskTop screen. You have your work disk in the drive (not the original GEOS disk.) Try clicking or double clicking on different icons and other areas of the screen. You cannot hurt anything, so enjoy yourself.

As you select various icons and try different commands, GEOS will often open a dialogue box in the middle of the screen telling you that there is insufficient room on the disk to activate that particular option.

The problem here is that GEOS IS a disk intensive operating system and it must often store information on disk rather than in the computers memory - even the information you are working with! So you need to make room on the disk by erasing a few files. We are going to be erasing a number of files from your work disk now, but don't worry about losing anything. Later, you will be making more work disks and tailoring them to your own needs.

To erase a file, simply drag the file's icon to the Waste Basket and drop it. That is, place the pointer on a file icon, click once, wait a moment, and click again. This will attach to the pointer a blue shadow of the original icon. You can drag the pointer and the blue shadow of the icon around the screen, placing it on the border, sending it to the printer, or dropping it in the Waste Basket.

This is a work disk, so you will not need the GEOS, GEOS BOOT, and GEOS KERNAL files. You only need these files on the original GEOS boot disk and on the security backups. These are only used when you first load GEOS each time you turn on your computer. So place the pointer on the GEOS icon, click once, wait a moment, click again, and drag the shadow down to the Waste Basket. With the shadow over the Waste Basket, click the fire button again to drop it.

Wait a minute...this doesn't work. GEOS will open a dialogue box in the middle of the screen telling you that "This file is write protected and can't be deleted." Berkeley Softworks made it difficult to erase GEOS, GEOS BOOT, and GEOS KERNAL by protecting them on the disk itself. This is a good idea because these three files are essential for loading GEOS; if you do not have these three files on your boot disk you will not be able to run GEOS. On a work disk, however, they only take up space.

It is still possible to erase these files, although you will have to go through a couple more steps. First, select the file by clicking once on the file icon. This will reverse the image on the deskTop pad. Then move the pointer to the command menu in the upper left of the screen, placing the pointer on 'file.' Click once to open the file command menu. Move the pointer down to the 'info' option, and click again. (We will occasionally use a form of shorthand to describe the selection of an option from the Command Menu. For example, this case might be described as selecting a file icon and then selecting File/info from the

Command Menu.) This will cause GEOS to open in the middle of the screen a dialogue window which contains information about the file. The only piece of information we need to worry about right now is the Write Protect status. The little box to the left of the words "Write Protect" will be filled in with black if the file is write protected on disk.

Several files on the original disk are locked, but for now we are concerned only with are GEOS, GEOS BOOT, GEOS KERNAL, an Backup. We want to erase these three files now, so select the icons for each of these three in turn, select the File/info option from the Command Menu, and click once on the Write Protect box for each. Once you have done this, you can go back and erase these files by dropping each of them in the Waste Basket.

Now we are getting somewhere. By erasing these three files, you have opened up about twenty-three thousand bytes (characters) on this work disk. This is still not a whole lot of room (this disk can hold about one hundred and seventy thousand bytes), but it is enough to start out.

If you need more space on the disk, you might want to erase some of the files that you will not be needing just yet, like the Photo Manager and the Preference Manager. You may want to use some of these later on another work disk, but for now they are not very useful.

Now you have a work disk, or rather a play disk. You will be able to make other work disks later. You are probably ready to work with GEOS on your own for a few minutes. It is important to move around to see what you can see and to find out what works. All the words in the world cannot replace a few minutes of experience, so explore. When you are ready, come back to the book.

deskTop

We have given a brief description of the deskTop screen, and terms for its features. There are a number of other terms we will be using throughout the book. Most of the terms are defined in the GEOS User's Guide, a few are not.

BITS & BYTES

The whole purpose of using a computer is to manipulate data. Computers handle data in terms of BITS and BYTES. A byte is roughly equivalent to one character, such as the letter "a," the numeral "3," or a punctuation mark such as "!." There are several other characters that never appear, but are used by the computer. You may have noticed that the deskTop notepad shows the space used and the space available on the disk in terms of bytes.

A byte contains eight bits of information, each of which may be either a one or a zero. This allows two hundred and fifty-six possible combinations of the eight bits that make up a byte. Thus, there are two hundred and fifty-six distinct bytes.

A byte may have several meanings depending on the context in which it is used by the computer. Usually, however, we need only be concerned with bytes that signify visible characters.

Bytes are counted in terms of Kilobytes. A Kilobyte, or Kbyte, is one thousand and twenty four bytes. Thus, the original GEOS disk has 166 K bytes, or almost 170,000 (166 x 1024 = 169,984), bytes of information on the disk.

DISK FILES

These bytes are not just all jumbled up together, but stored in disk files. A disk file is an ordered collection of bytes that can somehow be used by the computer, either as a program or as information used by programs.

Each disk file has a name and other information that is stored in the disk directory as a directory entry. The directory entry contains the name of the disk file, a file type indicator, the size of the file, and a pointer that indicates where on the disk the disk file begins. GEOS file directory entries contain additional information including the date and time stamp, an additional pointer which indicates where the icon and the information about that file are stored, and a secondary file type indicator used by GEOS to indicate how that file is used by GEOS.

The information in a disk file is actually stored in blocks (sectors) of 254 bytes. If you LOAD the directory of a disk from BASIC with a

```
LOAD "$",8 <RETURN>
(wait until the cursor appears, and then)
LIST <RETURN>
```

the computer will show you a list of the files that are on the disk. The number to the left of the file name tells the number of two hundred and fifty-four byte blocks that the file uses. The characters in the middle of the list are the names of the files as they are stored on the disk directory, and the three characters to the right of the filename give the standard type of the disk file. Here you may find PRG (program), SEQ (sequential), USR (user), and REL (relative) files. A less-than (<) sign to the right of the filetype indicates that the file is locked and cannot be changed without first unlocking the file (it is not obvious within GEOS that a file is locked - this only becomes evident when you open the information window for a file and see that the Write Protect box is filled in, or when you try to change a locked file by dropping it in the waste basket, or when you replace a document file). GEOS uses another byte, not visible in the directory entry, as a secondary file type identifier.

GEOS uses this secondary file type identifier to signify System Files, Application Files, Application Data Files, Desk Accessory Files, etc. Other special bytes on disk signify the class of the file, such as "Write Image" for a geoWrite document, or "Prfrnce Mgr V1.0" for the Preference Manager. Still other bytes stored on disk show the file structure, the name of the author of the file, the icon that represents that file on the deskTop notepad, and the comments that are given when the File/Info command is selected. The structure of files and directory entries is more fully explained in section two.

Most of these bytes that identify the type, class, structure, etc., of the file are stored in a single block pointed to by a secondary track/sector link within the directory entry.

PROGRAMS & DATA

The information stored on disk is of little use unless the computer can do something with it. So, there are two basic distinctions between files. On the one hand are the programs that control the activity of the computer; on the other hand are the data files that are created by the program files. GEOS, GEOS BOOT, GEOS KERNAL, BACKUP, geoPaint, geoWrite, notepad, text manager, and preference manager are all program files that contain the instructions that tell the computer how to do various things. The original GEOS disk has no data files.

You, as the user, must create the data files. The first data files that you will create will be geoWrite and geoPaint documents. A geoWrite document is represented on the deskTop notepad by an icon showing three pages, with the name of the disk file beneath. A geoPaint document is represented with a filled in frame and a brush, again with the name underneath.

Other data files that you may later create include Notes, Preferences, Text Scraps, Photo Scraps, Text Albums, and Photo Albums.

LAYOUT & FEATURES

The GEOS deskTop is the primary user interface of the GEOS system. From the deskTop, you can enter the GEOS applications; activate the deskTop accessories; copy, move, erase, and print files; exit to BASIC; and reset the system. In short, deskTop provides the greatest range of features within GEOS. It is the center of the GEOS system.

A short description of the GEOS deskTop was given on pages 5 and 6. What follows here is a more specific, more detailed explanation of the deskTop layout and features.

The GEOS deskTop has three major areas - the deskTop notepad, the Command Menu, and the border - which are further subdivided into several parts. The deskTop notepad is the most obvious part of the deskTop screen, occupying the upper left center part of the screen. Above and to the left of the deskTop notepad is the Command Menu. Everything else is, or lies within, the border.

The deskTop notepad displays, starting at the upper left:

disk name: one to 16 alphanumeric or symbol characters, except backarrow, british pound, pi, quote, dollar sign, commercial at, colon, semi-colon, asterisk, or the Commodore graphic characters shown on the front of the keys;

close button: placing the pointer on this box and pressing the fire button will close all the files to this disk, and prepare GEOS for having another disk inserted into the drive;

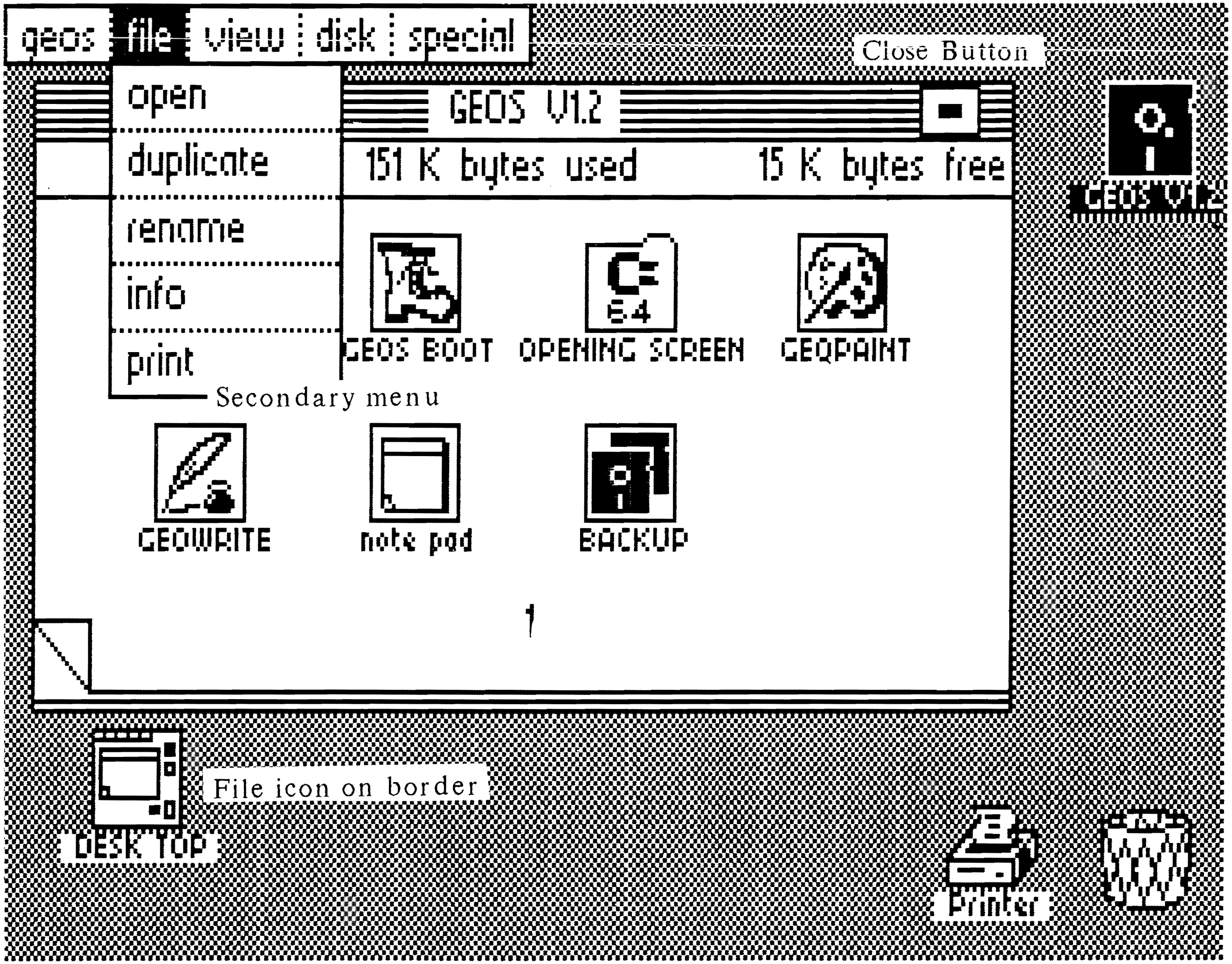
disk information: from left to right, the number of files on this disk, including the files in the border; the number of bytes used and the number of bytes available on the disk (166 Kbytes, or about 170,000 bytes total);

disk file icons: a sprite type graphic, plus characters, that represent the files on the currently active disk in the currently active drive. The icons appear in the order that the files appear on disk. In situations where GEOS defaults to the first file of a type on disk, the first may be determined by looking at the relative positions of the icons. Icons are ordered sequential from the upper left corner of page one, going across before going down;

page flip (page advance): selecting the upper right hand corner of the dog-earred page in the lower left of the deskTop notepad will advance the notepad to the next page, causing the on that page to be shown. Selecting the lower left corner of the dog-ear will flip the notepad backward, causing the files on the previous notepad page to be shown. Several such pages are available. (In theory, one hundred and forty-four files - eighteen pages - will fit in the standard directory of the disk, plus another eight files in the auxiliary directory, for a total of one hundred and fifty-two files. File data will normally fill the disk long before the directory is filled.);

page number: indicates the sequential page number of the deskTop notepad. Note, however, that if page 5 is not filled, GEOS will not advance to page 6.

The Command Menu bar in the upper left corner offers five options for controlling GEOS, activating some of its features, and for manipulating files. Selecting one of the Command Menu options will cause a secondary menu to drop down from the Command Menu bar. Selecting one of the options from these secondary menus will call a GEOS function, open a dialogue box, or provide information, depending on the menu item selected and the icons selected prior to choosing a primary menu option.



geos
file
view
disk
special
Close Button
open
duplicate
rename
info
print
Secondary menu
GEOS V1.2
151 K bytes used
15 K bytes free
GEOS BOOT
OPENING SCREEN
GEOPAINT
GEOWRITE
note pad
BACKUP
File icon on border
DESK TOP
Printer
GEOS V1.2

The GEOS command will, when selected from the Command Menu bar, display a drop down menu with a variable number of secondary menu items. The number depends, in part, on the number of deskTop accessories on the disk. For example, if the Calculator is on the disk it may be selected by first selecting GEOS on the Command Menu, then moving the pointer down to "calculator" and selecting that option. If the Calculator is not on the currently active disk, it will not be offered in this secondary menu. Some of the secondary commands of the GEOS menu will be available within GEOS applications such as geoWrite or geoPaint, others will not, as noted below. The GEOS commands include:

GEOS info: display credits and copyright information regarding GEOS. Not available within an application;

deskTop info: display credits and copyright information regarding deskTop. Not available within an application;

(*) info: display credits and copyright information regarding the currently resident application;

select printer: to inform GEOS of the printer which you have connected and will be using to print GEOS documents. GEOS defaults to the first printer driver on the boot disk. (The printer driver which appears first either in the GEOS deskTop notepad starting with the top left corner of page one, going first across the top line of each page; or first in a BASIC directory listing is the default. Removing all but the desired printer driver will force GEOS to default to the one you choose.) Not available within an application;

select input: similar to select printer, but for input drivers such as joystick or lightpen. Only the joystick driver is included with GEOS V1.2, but drivers for the Inkwell Flexidraw Lightpen (tm) and the Koala Technologies Koala Pad (tm) have been released via Quantum Link. Not available within an application;

note pad: activate the notepad, a desk accessory in which you can enter and save nine notepad pages of information. Available within an application;

preference manager: a deskTop accessory in which you can set the time and date for file stamping, alter the responsiveness of GEOS to the selected input device, alter the appearance of the pointer, and change the colors of the GEOS screen. Changes (border color, etc.) within the preference manager are effective for the current session only, unless the changes are saved to disk. Available within applications;

photo manager: a deskTop accessory by which you can create a photo album for the storage of photo scraps, and by which you can store or recall photo scraps. Available within applications;

text manager: a deskTop accessory by which you can create a text album for the storage of text scraps, and by which you can store or recall text scraps. Available within applications;

calculator: a deskTop accessory which provides four arithmetic functions, scientific notation, and eleven digits. The keyboard or the selected input device may be used to enter on the calculator. Available within applications;

alarm clock: a deskTop accessory by which you may set the time for an alarm to sound, and to set the time for date and time stamping. Available within applications.

The FILE command will, when selected after first selecting a file icon, display a drop down menu with the following commands:

open: run an application, deskTop accessory, or program outside of GEOS. Also, if you open an application data file, such as a geoWrite document, GEOS will automatically load geoWrite and then the document that you "opened."

duplicate: GEOS will open a dialogue box requesting you for a name for a duplicate copy of the selected file, and to create a duplicate copy of a file with the name you give on the same disk. The original file remains intact;

rename: GEOS will open a dialogue box requesting you for a new name for the same file. This changes the name of the file on disk, but otherwise leaves the file intact;

info: GEOS will read from the disk the information which it has stored about the selected file. This information includes the name of the file, the type, class, structure, size, date and time of last modification, author, write protect (lock) status, and optional comments;

print: GEOS will translate into bitmap graphics for the currently selected printer the data contained in the currently selected application (geoWrite or geoPaint) file and send this data to device number 4 (default device number for printers) on the serial bus.

Selecting the VIEW command will cause GEOS to display options for the format in which the disk directory will be displayed. Files may be selected (clicked on) only when the files are represented by icons. Also, when file information is viewed other than as icons, the file information is not presented on pages, but rather within a window onto a sequential list. Options for viewing include:

by icon: the default. Disk files are represented by icons shown on the deskTop notepad;

by size: disk file names, sizes, and type will be displayed as text on the deskTop notepad, with the largest file appearing first;

by type: disk file names, sizes, and types are displayed alphabetically according to the type of the file. No secondary sort is done on the file names or the file sizes, so multiple files of the same type will appear in the order in which they are entered in the disk directory;

by date: disk file names, sizes, and date and time of last modification are displayed in the order of the date of last modification, with the most recently modified file displayed first. Again, no secondary sort is done;

by name: disk file names, sizes, and types are displayed in alphabetical order, with symbols and numbers preceding uppercase letters, and uppercase letters preceding lowercase letters.

The DISK command will, when selected from the Command Menu, drop down a menu which offers commands for controlling the currently selected disk and GEOS's communication with the currently selected disk drive. Options include:

open: select (initialize and read part of the directory information of) the disk in the currently selected drive. This command is to be used when a data disk has been inserted in the disk drive, particularly when the disk icon at the right displays a question mark. Selecting Disk/open is functionally similar to selecting the disk icon;

close: deselect the disk in the currently active drive. This informs and prepares GEOS for a change of data or program disks and is good practice, especially if you are not careful about renaming your disks. Selecting Disk/close is functionally identical to selecting the Close Disk box in the upper right corner of the deskTop notepad;

rename: GEOS will open a dialogue box which prompts you for a new disk name, which it will then write to the disk. It is important to use a different name for each disk because GEOS uses the disk name to identify the disk when it writes information to the disk;

copy: GEOS will copy all the files from one (the source) disk to another (the target) disk. The copy function tests for a one or two drive system. If it is run on a one drive system, the copy function will prompt you to swap the source and the target disks. If it is run on a two drive system, the copy function will automatically copy the disk in the currently selected drive (device 8 or device 9) to the disk in the other drive (device 9 or device 8). If the target disk is already formatted as a Commodore disk, GEOS will ask for confirmation before proceeding with the copy function. If the target disk is not already formatted as a Commodore disk, GEOS will first prompt for a disk name and format the disk. This is a very slow way to copy on a single drive system;

add drive: this command allows the use of two disk drives both of which are device number 8. If you wish to use two disk drives, both device number 8, you must either turn on one drive, boot GEOS, select the Add Drive option from the disk menu, and follow the prompts; or turn on one drive, change this one drive to device number 9 with the following lines (from BASIC):

```
OPEN1,8,15,"M-W"+CHR$(119)+CHR$(0)+CHR$(2)+
CHR$(41)+CHR$(73) <RETURN>

CLOSE1 <RETURN>;
```

turn on the other disk drive (device number 8), and then boot GEOS from either disk drive;

validate: selecting this command will cause GEOS to reconstruct the directory of the selected disk. This option should be used when you are in doubt about the accuracy of the directory, after doing a standard Commodore DOS validate (V0:), after doing a standard Commodore DOS scratch (S0:), and after using non-GEOS programs that writes information to the disk. The directory contains information essential to the finding of data on the disk. If the directory is corrupt, the trouble may spread and make the data on the disk irretrievable;

format: selecting this command will cause GEOS to prompt you to place a disk in the currently selected drive and to give a name for the disk. GEOS will then prepare (format) the disk to store data. When a disk is formatted using this function, the disk will be prepared as a GEOS disk, including the extra information in track 18 sector 0, and the extra directory block which GEOS uses for the border.

The SPECIAL command will, when selected from the Command Menu, drop down a submenu with commands for non-standard GEOS functions. These include:

BASIC: selecting this command will cause GEOS to execute a system warm start, returning the computer to the BASIC environment/operating system, and the drive to standard Commodore DOS. The system warm start does NOT return the system to the original power-up state, but very close to it. Most BASIC and machine language programs will run without difficulty. In either case, you may return to GEOS by striking the <RESTORE> key. This is because the system warm start leaves the Non-maskable Interrupt (NMI) intact. This NMI points to the GEOS BOOT routine that resides at $C000 (decimal 49152);

RESET: selecting this command will cause GEOS to execute an internal warm start, re-initialize the disk drive, and return to GEOS;

Q-LINK: selecting this command will cause GEOS to prompt you for a Quantum Link system disk, and then load and execute the Q-Link telecommunications software.

Undocumented deskTop Features
and other observations

The directory entries for the files that are represented by icons on the deskTop notepad are held in a nearly standard directory on track 18. Standard Commodore DOS directory entries begin on track 18, sector one, and progress via the track and sector link at the beginning of each directory sector, to track 18, sectors 1, 4, 7, 10, etc., using every third sector in succession. A GEOS disk directory begins at track 18, sector 1, but then progresses every eighth sector in succession - 1, 9, 17, 6, etc. This will be invisible to the user.

The directory entries for the files represented on one deskTop notepad page are all stored in one directory sector. For example, all the directory entries for the files represented by icons on page one - GEOS, GEOS BOOT, GEOS KERNAL, geoPaint, geoWrite, note pad, and BACKUP - are all stored on track 18 sector 1. Each directory sector can hold directory entries for eight files, and the auxiliary directory block that is used to hold the directory entries for the files in the border can hold no more than eight directory entries. Thus the limit of eight files per notepad page, and eight files in the border for each disk.

GEOS will always use the first available blank directory entry space for the directory entries. This has the effect that you will not be allowed to use page 5 of the deskTop notepad until all the previous pages, up to page 4, are filled. Likewise, if you create enough files that you have more than thirty-two files on disk, thus forcing a fifth directory sector, there will be a page five to the deskTop notepad.

Another feature, however, can be troublesome. The directory entries for the files that are represented by icons in the deskTop border are not contained within the directory blocks on track 18. Rather, the directory entries for these files are contained in an auxiliary directory block, usually found at track 19, sector 8.

This can be tested by listing a standard directory from Commodore BASIC. First, note the names that appear with the icons in the deskTop border. Note also the total number of files appearing on the pages of the deskTop notepad. Then go to BASIC, LOAD and LIST the directory. The names of the files that are represented by icons on the border will not appear.

Another test is to count the blocks used by the files listed in the standard directory and compare this with the sum (664-blocks free) which gives the actual number of blocks used. If a file icon appears in the boarder of the deskTop screen, these two numbers will not agree.

This undocumented feature can corrupt the disk in at least one case. If a file on the border of the deskTop screen is locked and you attempt to unlock the file and then erase it by placing it in the Waste Basket, it may not work. You will have changed the information regarding the disk that is stored in the computer, but not the information on the disk. Information inside the computer about which blocks on the disk are available can be incorrect.

To test this, take a fresh work disk (a copy of the original GEOS disk which has not had any files erased or changed), and try to erase the deskTop file from the border. First move the deskTop icon to the border. Then click on the icon, select the File/info command, click on the Write Protect box, and then drag the deskTop icon to the Waste Basket and release it. The deskTop icon will usually disappear. Now click on the disk icon to re-initialize the disk. The deskTop icon will reappear. Selecting File/info will show that the deskTop file is still locked. This does not always work, and the deskTop file will, on occasion, be erased.

Another effect of this corruption of the map of the disk that GEOS holds in computer memory can be seen by dragging the Backup icon to the border, selecting File/info, changing the Write Protect status, and then clicking once in the border. The Backup icon will look rather odd. Now, select the Backup icon again, and select File/info. You will usually get the information for another file, such as deskTop or GEOS BOOT. This effect will not always occur, but it can corrupt the actual data on the disk when it does.

A number of other features of deskTop can be confusing. For example, the GEOS User's Guide states that the GEOS will default to the first printer driver and the first input driver on the disk. It fails to mention, however, that it is the first printer driver and the first input driver on the BOOT disk, not on a work disk. Thus, unless you happen to use an Commodore MPS 1000 or compatible printer, you must either use the GEOS/select printer option every time you load GEOS, or you must move files around on your boot disk. If your current work disk does not have the default printer driver, as determined by the order of the printer drivers on the boot disk, GEOS will give you an "Error reading disk."

Generally, it is not a good idea to move files around on the original GEOS disk. It should not hurt to remove the write protect tabs and to move the position of the files, but in doing so there is always some level of unnecessary risk, and so cannot be recommended. Rather, use one of the utilities in section two to create another boot disk and store your original GEOS disk away. Then you may rearrange the order of printer drivers and input drivers on your home-made boot disk without fear of corrupting the original.

Another limitation of GEOS is the number of files that it can actually work with at any one time. The problem is not with the total number of files, but rather with the number of files of each type that it can work with. For example, the GEOS command can only show the first eight deskTop accessories in the order that they appear on the disk. If you have more than eight deskTop accessories (which is not yet possible, as there are not yet that many available) on your disk, and the one you want to use appears in the ninth position, you cannot access it from the GEOS menu. If you are in deskTop as opposed to an application, however, you should be able to select deskTop accessories by double-clicking on the appropriate icon, even if the accessory is not one of the first eight on the disk.

Likewise, geoWrite and geoPaint are limited in the number of document files that they can show on the menu within the application. When you open either of the applications from deskTop, or when you close a current document and return to the opening application menu, and choose to "Open existing document," only the first fifteen application data files will be shown on the directory listing. If you wish to open a currently existing document from within geoWrite or geoPaint, it must appear as one of the first fifteen documents on the disk for that application. You can, however, open a document not in the first fifteen directly from deskTop by double-clicking on the file icon or by clicking once and selecting File/open.

Once you have booted GEOS, you may swap disks freely without concern for whether or not deskTop is on each disk - until you load an application program such as geoWrite or geoPaint. Application programs replace deskTop in the computer's memory. Upon exiting an application, the GEOS operating system looks for deskTop to be where it was first loaded from. If it is not there, you must replace the disk with one that has deskTop on it before you proceed. About one time in twenty, GEOS will crash if deskTop is not present when you leave an application. This is no more than an inconvenience, as all data will have already been saved. You must simply reboot.

Interestingly, GEOS apparently can be booted from a drive of almost any normal device number, from device 8 through device 11. It cannot, however, address device number 10 or 11 once it has been booted. For example, I have attached three 1571 disk drives to my computer, device 8, 9, and 11. I LOADed "GEOS",11,1. It loaded and ran perfectly - as long a disk in one of the other two drives had the deskTop file on it. Indications are that GEOS might support more than two drives, although the support is either not fully developed or simply not implemented.

Although it does not deal directly with deskTop, it is interesting to note that the order of the first few files on the disk makes a difference. It is easiest to load GEOS with the command

```
LOAD ":*",8,1 <RETURN>
```

This assumes that the file "GEOS" is the first file on the disk. For most programs, the order of the files on the disk does not matter all that much, provided the copy protection is not too heavy. With GEOS, however, the order of the files on the disks can make a big difference. The GEOS KERNAL must be the third file in the directory, and appear as the third icon on the first deskTop notepad page. If the GEOS KERNAL file is not in the third position, the GEOS BOOT program will crash and the disk drive read/write head might get stuck. If this happens - if the computer cannot seem to access the drive, and you cannot even load a directory - send the following command to the drive:

```
OPEN15,8,15,"I0:":CLOSE15
```

This will initialize the drive, and should get the read/write head unstuck. If problems continue, please refer to the appendix on Erros & Problems.

Here, the need to have the GEOS KERNAL file in the third position on the disk has nothing to do with the copy protection, but rather with the Turbo disk routines.

And, finally, because the files that are represented by icons that are on the border of the deskTop screen have their directory entries in the GEOS auxiliary directory sector, you will not be able to load these files from outside of BASIC. For example, if you have a BASIC program called "Random Numbers" on a GEOS disk, and you put its icon on the border of the deskTop screen, you will not be able to load "Random Numbers" from BASIC. It can only be loaded from within GEOS, unless you put the icon back on a deskTop notepad page.

geoWrite

GeoWrite may be considered either a text editor or simple word processor, but it does not fit neatly into either category. Rather, it provides for the relatively free form entry of text onto a graphic depiction of a document page. The text may be shown, and later printed, in variety of proportionally spaced fonts including different typestyles (BSW, University, Cory, Dwinelle, etc.), enhancements (bold, italic, outlined, or underlined), and sizes. GeoWrite also allows graphics to be included with the text on screen and when printed.

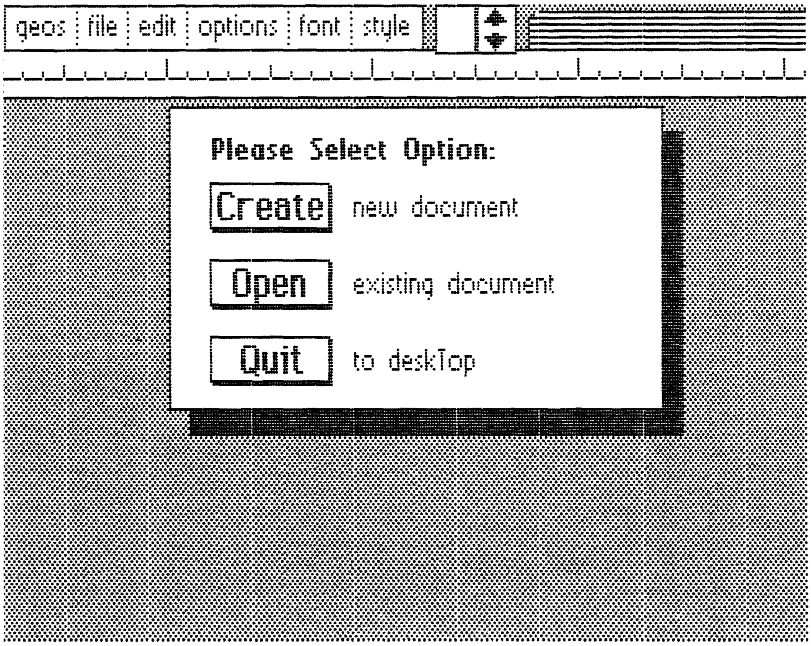
This is not to say the geoWrite is ideal. As with every other computer program, it suffers from its advantages. Nothing is free, and every feature takes its toll in other features that might have been offered. One of the first handicaps you will notice in geoWrite is the lack of speed. Its advantages - the ability to use multiple proportionally spaced fonts, graphics, and to see on the screen almost exactly what you will get on paper - are unmatched on the Commodore VIC series of computers. Although geoWrite may not be suitable for long term papers or books, it is an excellent utility for producing a few attractive pages of text and graphics.

As word processors go, geoWrite is relatively easy to use. The entry, enhancement, and formatting of text is quite straightforward. The simplicity is further aided by the pointer/menu command structure. To enter text, simply type. To enhance text, select Style on the Command menu and then click on the enhancement you want. But we are ahead of ourselves. First, let us describe how to get into geoWrite.

From the GEOS deskTop, the first time you use geowrite you must open the program file by either double clicking on the geoWrite icon, or by clicking once on the geoWrite icon and then selecting the File/open command. Later, after you have one or more documents already created on the GEOS disk, you may run geoWrite from deskTop by selecting File/open or double clicking on any geoWrite document.

For example, you have used geoWrite to create a document named "letter 1." You may enter geoWrite and edit "letter 1" by double clicking on the "letter 1" icon, or clicking once and selecting the 'open' option on the File sub-menu. Either way, you will enter geoWrite, and geoWrite will automatically load the "letter 1" document.

Any time you enter geoWrite by selecting the geoWrite icon from deskTop, the geoWrite application program will load and present you with the following screen.



Please Select Option:

Create new document

Open existing document

Quit to deskTop

Since this is the first time you are using geoWrite, choose the "Create new document" option. GeoWrite will then ask you to give a name for your new document, create a file on the disk with that name, and will then be ready for you to enter the text for your new document.

If you want to edit a document that you have worked on before, select the "Open existing document" option. This will present you with yet another menu. This menu gives you a list of the geoWrite documents on the current disk. The menu window only shows the names of five files at a time, but you can click on the up and down arrows to show more files. This menu will not, however, show more than the first fifteen geoWrite files on the current disk. If you have more than fifteen geoWrite files on the disk, and you need to work with one that appears, say, in the seventeenth position, you will need to either rearrange the files on the deskTop notepad, or you can open the seventeenth geoWrite file from deskTop.

If you are using two disk drives (devices 8 and 9), a third option will be presented on this menu. The "Disk" option will allow you to open an existing document that is stored on a disk in the other disk drive. Taking advantage of this feature will be explained more fully under Hints & Tips on GeoWrite.

When you first enter geoWrite, the program defaults to using the standard BSW 9 point font. If you wish to use any other fonts, the Font file must be present on the disk. If you are using one drive, it must be present on the disk with geoWrite; if you are using two drives, the fonts may be present on the disk in either of the drives.

If you have the Berkeley Softworks FontPackl (tm), you will not be able to use all the fonts in the same document. The FontPack simply provides different shapes for the characters. To use the FontPack fonts, you need to copy the files on the FontPack disk to your GEOS work disk, and then make sure that all the fonts you will be using are among the first eight fonts that appear on the disk.

USING GEOWRITE

Layout & Features

The geoWrite screen is in many ways similar to the deskTop screen. The main difference is that geoWrite presents you with a window onto a document page that takes up most of the screen in place of the deskTop notepad. This window shows a small part of a document page.

First, of course, is the pointer. The pointer is common to all GEOS applications, accessories, and functions except BACKUP. In addition to selecting options on the command bar, the pointer is used within geoWrite to move around and within the document. You may position the cursor bar by pointing at a location on the page; you may not, however, position the cursor bar anywhere beyond the active text area. If you have just entered geoWrite, chosen to create a new document, and have not entered any text, you will not be able to move the cursor bar from the upper left of the page. No matter where you point and click, the cursor bar will remain in the first position. After you have entered a bit of text, you may then use the pointer to position the cursor bar anywhere within the range of the text that you have entered.

Directly above the document page is a ruler line that shows the position of the text on the document page with respect to the left margin of the document page as it will be printed. The ruler line is divided into two unequal parts, the lower part showing the positions of the left and right margin; the upper part showing the horizontal length, in inches, of the part of the page you are viewing. The window does not actually show the complete document page as geoWrite is constrained to use a hard left margin that is one and two-tenths inches from the left edge of the printed page, and a right margin that is one and one-half inches from the right edge of the printed page. (This assumes that you are using an 80 dot per inch printer.)

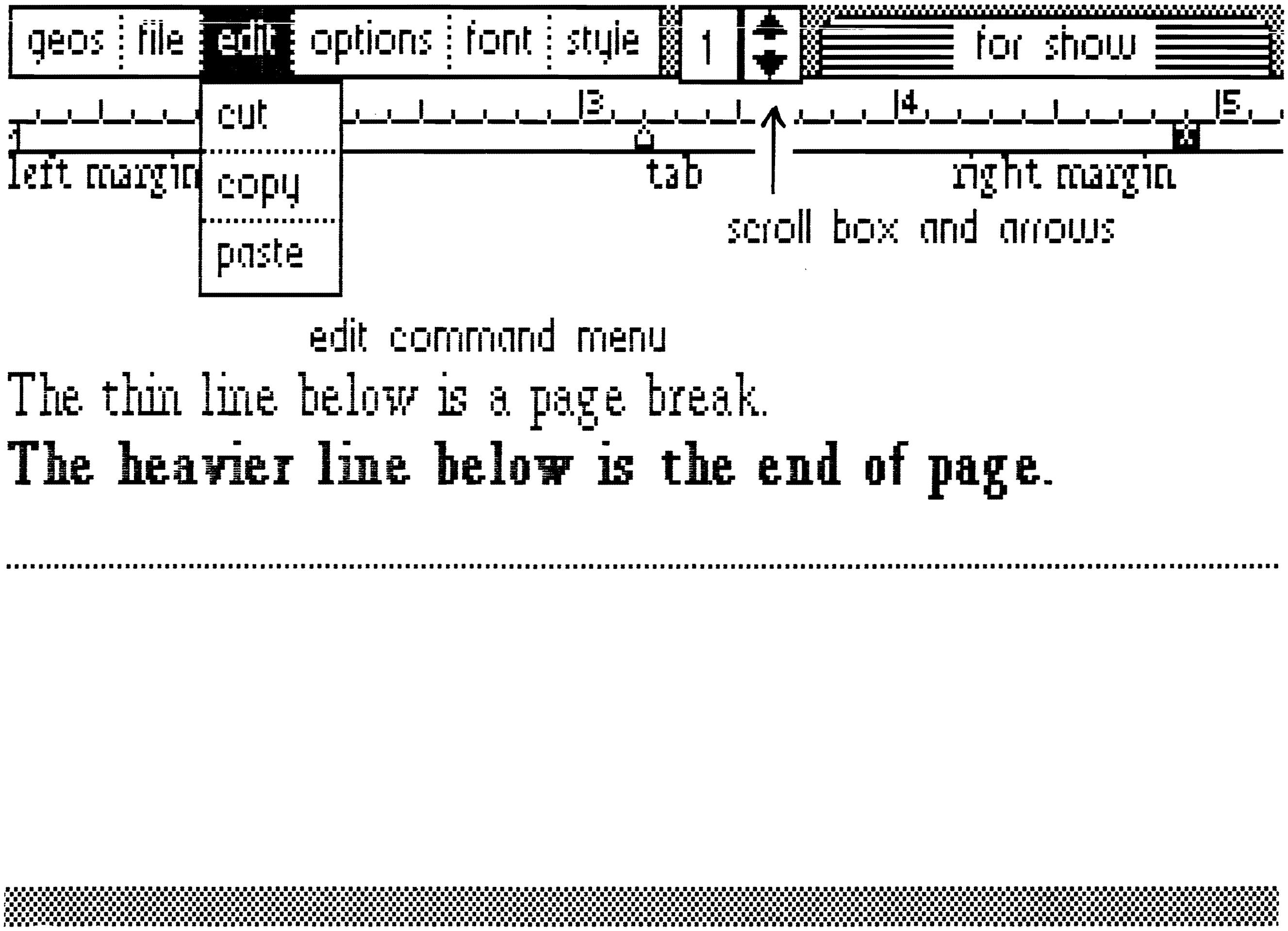
In the upper right corner of the screen, geoWrite displays the name of the document that you are currently working on in the Title Bar which appears like the tab of a file folder.

To the left of the Title Bar are the Scroll Arrows and the Scroll Box/Page Indicator. By placing the pointer on either the up or down Scroll Arrows, you can cause geoWrite to move the window one line in either direction against the background of the document page. The window will move one line for each click, or will continue moving in one line increments if you keep the fire button depressed.

The Scroll Box lies on top of the Page Indicator. Here, a little black rectangle indicates the relative position of the text window on the document page. If the black rectangle is in the extreme upper left of the page indicator box, then you are at the very beginning of the page. If the black rectangle is halfway down and on the right two-thirds of the page indicator box, then you are about midway through the page. The page indicator box also displays which page you are viewing of the document currently loaded. A document may be sixty-four pages long, otherwise limited only by disk space.

You can drag the black rectangle around the Scroll Box/Page Indicator to position your window on to the geoWrite document page by placing the arrow pointer in the box, clicking once, and then dragging the rectangle to the relative position within the box that corresponds to where on the document page you want your window to be. Click once more to release the position indicator, then move the arrow pointer onto the document window, and click.

An implicit function allows you to move the text window from the left two-thirds of the document window to the right two-thirds of the window. You may either move the cursor by entering text, or you may simply move the pointer. Moving either of these will cause geoWrite to move the window to the right two-thirds of the document page.



The thin line below is a page break.

The heavier line below is the end of page.

COMMAND MENU

Finally, in the upper left corner of the screen is the Command Menu. This Command Menu is very similar to the Command Menu of the deskTop screen, but has some different sub-menus, and some different options within some of the sub-menus that remain.

At the extreme left is the GEOS sub-menu. The options here provide information about geoWrite and allow you to use any of the first eight Desk Accessories on the currently selected disk. Options not available here, but available from deskTop, are "select printer," and "select input."

The FILE menu provides several options different from those offered in deskTop.

close: selecting this option will save the currently loaded document to the disk, and return you to the first geoWrite menu (Create/Open/Quit);

update: selecting this option will save the currently loaded document to the disk and return you to the document. This option is useful when you are working with a particularly long document, have made several changes to the document, or are about to try something with the document and are not sure how it will turn out;

preview: selecting this option will cause geoWrite to show you a graphic depiction of the complete document page as it will appear when printed. The Commodore 64 does not have sufficient resolution to produce such small text legibly, so the page shown by the preview option will be unreadable. It will, however, give you a good idea of how the text will appear, prior to printing. This option will show only the current page. To preview a multi-page document, you must move from one page to the next, and select the File/preview option for each page;

recover: selecting this option will, in most cases, allow you to recover from the changes you have made to the current page since the last time you saved the current page. Recover will NOT work if you have moved from one page to another. This option is particularly useful if you have cut a large piece of text from the page, and wish to have it back. If you attempt to recover, and a window opens telling you that geoWrite "Cannot recover from file," try again. Many times, the recover option will work on the second attempt. The recover option, however, is not wholly reliable. If the File/recover option does not work, you might want to try the Edit/paste/text option to recover lost text. The File/recover option can also be used to remove additions made since the last time the document page was saved;

rename: selecting this option will cause geoWrite to prompt you for a new name for the currently loaded document, and then to rename the document in the computer and on disk;

print: selecting this option will cause geoWrite to translate the currently loaded document, using the selected printer driver's translation table, and send the interpreted data to the printer;

quit: selecting this option will cause GEOS to save the currently loaded document as is to the disk from which the file was loaded. If the deskTop program file is on the disk, then GEOS will return to the deskTop. If deskTop is not on the disk, then GEOS will prompt for a disk that has deskTop on it. Sometimes, however, GEOS will go away and take your computer with it. If this should happen, the file will be intact on disk as it was in the computer when you selected the File/quit option, but you will have to reset your computer and reload GEOS.

Next is the EDIT menu. This menu provides options for pasting text and graphics, and for cutting text. Specifically,

cut: selecting this option will cause geoWrite to delete the selected area of text from the current document, and to place the deleted text into the Text Scrap file on disk, thus replacing any material that may have previously been placed in the Text Scrap on disk by the Edit/cut or the Edit/copy commands. If the Edit/cut option is selected, with no range of text selected, the Text Scrap file on disk will not be replaced and the document will not be affected;

copy: selecting this option will cause geoWrite to copy the selected range of text from the current document into the Text Scrap file on disk. The document itself will not be affected. If the Edit/copy option is selected with no range of text selected, the Text Scrap file on disk will not be affected. Text copied with the Edit/copy option will replace the contents of the Text Scrap file on disk;

paste: selecting this option will cause a tertiary menu to appear, allowing you to select either text or picture.

text: selecting this option will cause geoWrite to copy the text from the Text Scrap file on disk to the cursor position in the current document. The Text Scrap file on disk remains unaffected;

picture: selecting this option will cause geoWrite to copy the graphic data from the Photo Scrap file on disk to the cursor position in the current document. The Photo Scrap file on disk remains unaffected;

The OPTIONS menu provides five commands namely:

previous page: selecting this command will cause geoWrite to save to disk the text and graphics data on the current page and to load from disk the text and graphics data for the previous document page, and to display the top of the previous document page in the geoWrite text window. Selecting the File/recover option from the command menu will not work properly after moving from one page to another;

next page: selecting this command will cause geoWrite to save to the disk the text and graphics data of the current document page, and to load from the disk the text and graphics data for the next document page, and to display the top portion of that next document page in the geoWrite window. Selecting the File/recover option from the command menu will not work properly after moving from one page to another;

goto page: selecting this command will cause geoWrite to query you for the page you would like to view, and then to save the text and graphics data of the current document page to disk, and to load from the disk the text and graphics data for the chosen document page, and to display the top portion of that chosen document page in the geoWrite window. Selecting the File/recover option from the command menu will not work properly after moving from one page to another;

hide pictures/show pictures: (toggle) selecting the Options/hide pictures command will cause geoWrite to mask out with the deskTop border color the graphics data within the geoWrite document. The graphics data are maintained within the computer, and when the document is saved to disk, even with the graphics hidden. The Options/show pictures option will cause geoWrite to display the graphics data on the computer screen. The hide pictures/show pictures status is saved with the document file when the file is closed;

the geoWrite documents on the current disk. The menu window only shows the names of five files at a time, but you can click on the up and down arrows to show more files. This menu will not, however, show more than the first fifteen geoWrite files on the current disk. If you have more than fifteen geoWrite files on the disk, and you need to work with one that appears, say, in the seventeenth position, you will need to either rearrange the files on the deskTop notepad, or you can open the seventeenth geoWrite file from deskTop.

If you are using two disk drives (devices 8 and 9), a third option will be presented on this menu. The "Disk" option will allow you to open an existing document that is stored on a disk in the other disk drive. Taking advantage of this feature will be explained more fully under Hints & Tips on GeoWrite.

When you first enter geoWrite, the program defaults to using the standard BSW 9 point font. If you wish to use any other fonts, the Font file must be present on the disk. If you are using one drive, it must be present on the disk with geoWrite; if you are using two drives, the fonts may be present on the disk in either of the drives.

If you have the Berkeley Softworks FontPackl (tm), you will not be able to use all the fonts in the same document. The FontPack simply provides different shapes for the characters. To use the FontPack fonts, you need to copy the files on the FontPack disk to your GEOS work disk, and then make sure that all the fonts you will be using are among the first eight fonts that appear on the disk.

USING GEOWRITE

Layout & Features

The geoWrite screen is in many ways similar to the deskTop screen. The main difference is that geoWrite presents you with a window onto a document page that takes up most of the screen in place of the deskTop notepad. This window shows a small part of a document page.

First, of course, is the pointer. The pointer is common to all GEOS applications, accessories, and functions except BACKUP. In addition to selecting options on the command bar, the pointer is used within geoWrite to move around and within the document. You may position the cursor bar by pointing at a location on the page; you may not, however, position the cursor bar anywhere beyond the active text area. If you have just entered geoWrite, chosen to create a new document, and have not entered any text, you will not be able to move the cursor bar from the upper left of the page. No matter where you point and click, the cursor bar will remain in the first position. After you have entered a bit of text, you may then use the pointer to position the cursor bar anywhere within the range of the text that you have entered.

Directly above the document page is a ruler line that shows the position of the text on the document page with respect to the left margin of the document page as it will be printed. The ruler line is divided into two unequal parts, the lower part showing the positions of the left and right margin; the upper part showing the horizontal length, in inches, of the part of the page you are viewing. The window does not actually show the complete document page as geoWrite is constrained to use a hard left margin that is one and two-tenths inches from the left edge of the printed page, and a right margin that is one and one-half inches from the right edge of the printed page. (This assumes that you are using an 80 dot per inch printer.)

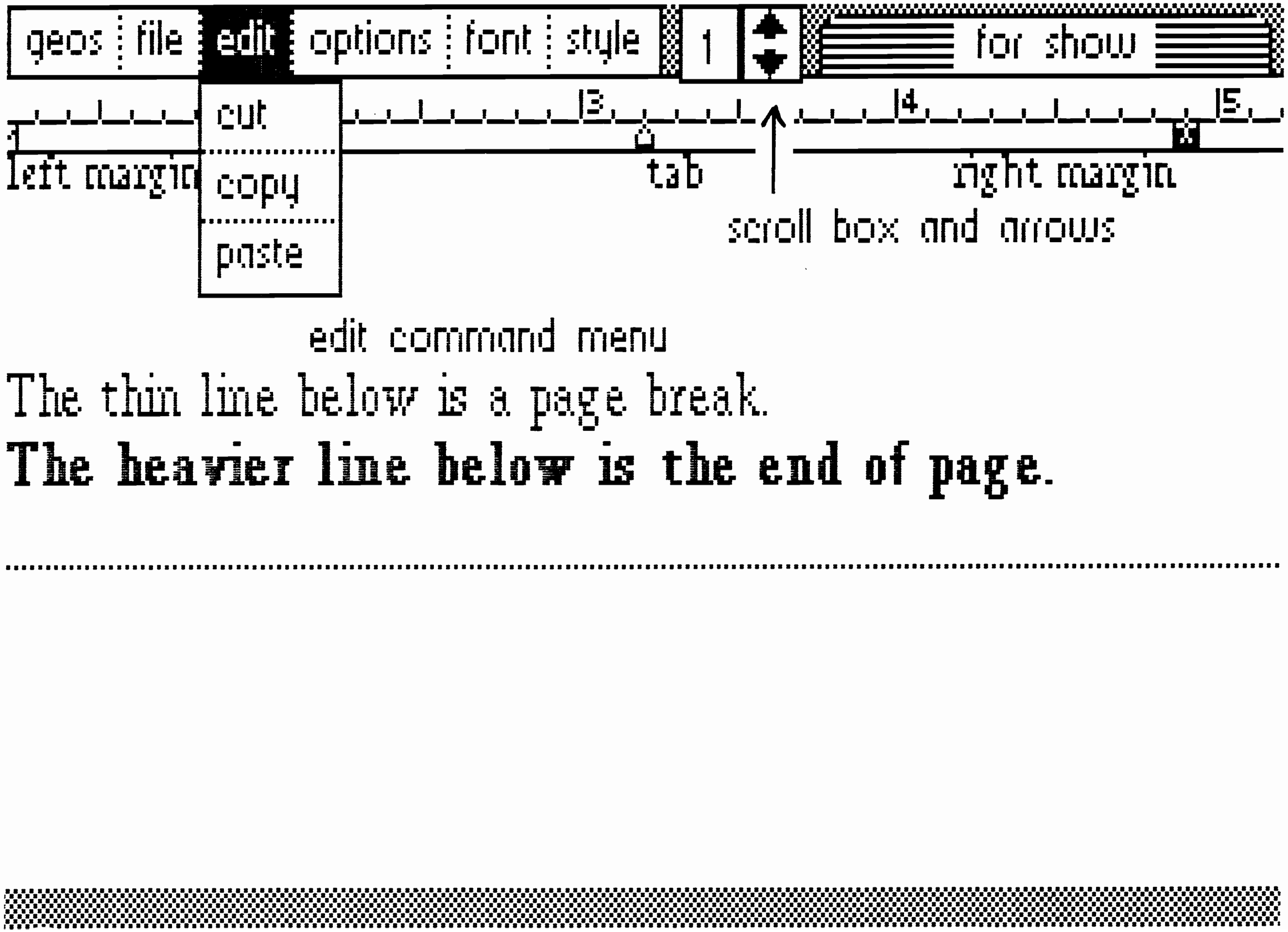
In the upper right corner of the screen, geoWrite displays the name of the document that you are currently working on in the Title Bar which appears like the tab of a file folder.

To the left of the Title Bar are the Scroll Arrows and the Scroll Box/Page Indicator. By placing the pointer on either the up or down Scroll Arrows, you can cause geoWrite to move the window one line in either direction against the background of the document page. The window will move one line for each click, or will continue moving in one line increments if you keep the fire button depressed.

The Scroll Box lies on top of the Page Indicator. Here, a little black rectangle indicates the relative position of the text window on the document page. If the black rectangle is in the extreme upper left of the page indicator box, then you are at the very beginning of the page. If the black rectangle is halfway down and on the right two-thirds of the page indicator box, then you are about midway through the page. The page indicator box also displays which page you are viewing of the document currently loaded. A document may be sixty-four pages long, otherwise limited only by disk space.

You can drag the black rectangle around the Scroll Box/Page Indicator to position your window on to the geoWrite document page by placing the arrow pointer in the box, clicking once, and then dragging the rectangle to the relative position within the box that corresponds to where on the document page you want your window to be. Click once more to release the position indicator, then move the arrow pointer onto the document window, and click.

An implicit function allows you to move the text window from the left two-thirds of the document window to the right two-thirds of the window. You may either move the cursor by entering text, or you may simply move the pointer. Moving either of these will cause geoWrite to move the window to the right two-thirds of the document page.



The thin line below is a page break.

The heavier line below is the end of page.

COMMAND MENU

Finally, in the upper left corner of the screen is the Command Menu. This Command Menu is very similar to the Command Menu of the deskTop screen, but has some different sub-menus, and some different options within some of the sub-menus that remain.

At the extreme left is the GEOS sub-menu. The options here provide information about geoWrite and allow you to use any of the first eight Desk Accessories on the currently selected disk. Options not available here, but available from deskTop, are "select printer," and "select input."

The FILE menu provides several options different from those offered in deskTop.

close: selecting this option will save the currently loaded document to the disk, and return you to the first geoWrite menu (Create/Open/Quit);

update: selecting this option will save the currently loaded document to the disk and return you to the document. This option is useful when you are working with a particularly long document, have made several changes to the document, or are about to try something with the document and are not sure how it will turn out;

preview: selecting this option will cause geoWrite to show you a graphic depiction of the complete document page as it will appear when printed. The Commodore 64 does not have sufficient resolution to produce such small text legibly, so the page shown by the preview option will be unreadable. It will, however, give you a good idea of how the text will appear, prior to printing. This option will show only the current page. To preview a multi-page document, you must move from one page to the next, and select the File/preview option for each page;

recover: selecting this option will, in most cases, allow you to recover from the changes you have made to the current page since the last time you saved the current page. Recover will NOT work if you have moved from one page to another. This option is particularly useful if you have cut a large piece of text from the page, and wish to have it back. If you attempt to recover, and a window opens telling you that geoWrite "Cannot recover from file," try again. Many times, the recover option will work on the second attempt. The recover option, however, is not wholly reliable. If the File/recover option does not work, you might want to try the Edit/paste/text option to recover lost text. The File/recover option can also be used to remove additions made since the last time the document page was saved;

rename: selecting this option will cause geoWrite to prompt you for a new name for the currently loaded document, and then to rename the document in the computer and on disk;

print: selecting this option will cause geoWrite to translate the currently loaded document, using the selected printer driver's translation table, and send the interpreted data to the printer;

quit: selecting this option will cause GEOS to save the currently loaded document as is to the disk from which the file was loaded. If the deskTop program file is on the disk, then GEOS will return to the deskTop. If deskTop is not on the disk, then GEOS will prompt for a disk that has deskTop on it. Sometimes, however, GEOS will go away and take your computer with it. If this should happen, the file will be intact on disk as it was in the computer when you selected the File/quit option, but you will have to reset your computer and reload GEOS.

Next is the EDIT menu. This menu provides options for pasting text and graphics, and for cutting text. Specifically,

cut: selecting this option will cause geoWrite to delete the selected area of text from the current document, and to place the deleted text into the Text Scrap file on disk, thus replacing any material that may have previously been placed in the Text Scrap on disk by the Edit/cut or the Edit/copy commands. If the Edit/cut option is selected, with no range of text selected, the Text Scrap file on disk will not be replaced and the document will not be affected;

copy: selecting this option will cause geoWrite to copy the selected range of text from the current document into the Text Scrap file on disk. The document itself will not be affected. If the Edit/copy option is selected with no range of text selected, the Text Scrap file on disk will not be affected. Text copied with the Edit/copy option will replace the contents of the Text Scrap file on disk;

paste: selecting this option will cause a tertiary menu to appear, allowing you to select either text or picture.

text: selecting this option will cause geoWrite to copy the text from the Text Scrap file on disk to the cursor position in the current document. The Text Scrap file on disk remains unaffected;

picture: selecting this option will cause geoWrite to copy the graphic data from the Photo Scrap file on disk to the cursor position in the current document. The Photo Scrap file on disk remains unaffected;

The OPTIONS menu provides five commands namely:

previous page: selecting this command will cause geoWrite to save to disk the text and graphics data on the current page and to load from disk the text and graphics data for the previous document page, and to display the top of the previous document page in the geoWrite text window. Selecting the File/recover option from the command menu will not work properly after moving from one page to another;

next page: selecting this command will cause geoWrite to save to the disk the text and graphics data of the current document page, and to load from the disk the text and graphics data for the next document page, and to display the top portion of that next document page in the geoWrite window. Selecting the File/recover option from the command menu will not work properly after moving from one page to another;

goto page: selecting this command will cause geoWrite to query you for the page you would like to view, and then to save the text and graphics data of the current document page to disk, and to load from the disk the text and graphics data for the chosen document page, and to display the top portion of that chosen document page in the geoWrite window. Selecting the File/recover option from the command menu will not work properly after moving from one page to another;

hide pictures/show pictures: (toggle) selecting the Options/hide pictures command will cause geoWrite to mask out with the deskTop border color the graphics data within the geoWrite document. The graphics data are maintained within the computer, and when the document is saved to disk, even with the graphics hidden. The Options/show pictures option will cause geoWrite to display the graphics data on the computer screen. The hide pictures/show pictures status is saved with the document file when the file is closed;

page break: selecting this command will cause geoWrite to force the insertion of a page break [chr$(12)] into the geoWrite document at the current position of the cursor. A page break character may be deleted in the same manner as any other character by placing the cursor in the upper left corner of the page following the page break and hitting the <DELETE>, ^h (control h), or left/right cursor key.

The next two selections on the command menu, FONT and STYLE, provide for the control over the style and size of text characters as the characters will appear on the document page, both on the screen and when printed.

You may select a font prior to entering the text by selecting Font/(font)/(size). When a font is chosen in this manner, the text that is then entered will be shown in the chosen font - provided you do not first move the cursor into a range where another font is active.

You may also select a font or enhancement by selecting a range of text and then selecting the desired font or enhancement.

Selecting a font prior to entering text has the advantage of being able to see the text as it will appear as you enter it. Selecting fonts and enhancements after the text has already been entered has the advantage of being significantly quicker. Every font that is used, with the exception of BSW 9 point, it must be loaded from disk.

The selection of a font inserts into the document a series of characters which flag a change of font and indicate the specific font. These flag and specification characters cannot be deleted from within a geoWrite document. These flag characters are more fully explained in section three, under file structure.

ENTERING TEXT

You may enter text into an opened geoWrite document by simply striking keys. Text will be entered into the geoWrite document at the position of the slowly flashing blue vertical bar cursor. The cursor may be positioned anywhere within active document space - that is, anywhere up to and including the last entered carriage return. You cannot place the cursor beyond the last character or carriage return on any given page. The following keys will perform certain functions and move the cursor as indicated:

<RETURN>: will enter a hard carriage return [CHR$(13)] into the geoWrite document and move the cursor to the left edge of the next line;

<CONTROL>m (^m): same as <RETURN>;

<CONTROL>i (^i): will enter a tab character into the geoWrite document and move the cursor to the next tab marker, if one is set;

<INST/DEL>: will delete the character to the immediate left of the cursor and move the cursor into that position. This key can be used to delete imbedded tabs, hard carriage returns, and page breaks;

<CONTROL>h (^h): the same as <INST/DEL>;

<CRSR LFT/RGHT>: the same as <INST/DEL>.

GeoWrite will accept all characters of the upper and lower case alphabet, all numeric characters, and all punctuation with the exception of the left arrow key (upper left corner of the keyboard), and the English pound sign (immediately left of the <CLR/HOME> key). Commodore graphics characters (appearing on the front of the keys) cannot be used.

GeoWrite supports a number of non-standard characters. The Commodore up-arrow (immediately left of the <RESTORE> key) appears within geoWrite as a caret (^). Other supplemental characters may be accessed by holding down the Commodore logo key (at the lower left corner of the keyboard) and pressing character key. Specifically:

<logo> / = backslash (\)
<logo> : = left curly bracket ({)
<logo> ; = right curly bracket (})
<logo> @ = grave (‘)
<logo> * = tilde (˜)
<logo> ^ = vertical bar (pipe ¦) (here, ^=up arrow)
<logo> - = underline (_)

FONTS & ENHANCEMENTS

When you first begin with geoWrite, all the text you enter will be in the BSW 9 point font. If you wish to use another font, you must select the Font/(font)/(size) command, where (font) can be any of the first eight fonts on the disk. When you select a font, you will be presented with a tertiary menu from which you select the size. For example, if you wish to use the Roma 12 point font, you must move the pointer to the box in the Command Menu containing the word "font,", click, move the pointer to the secondary menu box that contains the word "Roma", click, and then move the pointer to the tertiary menu box that contains the words "12 point" and click again (Font/roma/12). If you move the pointer outside the box at any time, you will have to start again by selecting "font."

Similarly, geoWrite defaults to no enhancement of the text as you type it in. You may enhance the text by selecting commands from the Style menu in the same manner as you select fonts. For example, to underline text you select Style/underline. Unlike the fonts, however, you may use any combination of enhancements at the same time. Thus you can have, say, California 12 point printed out in underlined italic.

Also unlike the fonts, the enhancements are toggled. This means that you can select bold, outline, underline, and italic all at the same time and then turn them off one by one, instead of selecting Font/plain text and turning the others back on.

For example, you are typing along in University 10 point plain text and then turn on all of the style enhancements - bold, outline, underline, and italic. Then, after the first word, you want to turn off the underline. Instead of selecting Style/plain text and then selecting bold, outline, and italic again, you need only select Style/underline to turn underline off and leave the rest of the enhancements unchanged.

You may also change the fonts and enhancements of text that has already been entered. To do this, you must first set a range of text. Use the pointer to place the cursor bar at the beginning of the text you wish to change, then press and hold the fire button as you move the pointer to the end of the text that you wish to change. This will cause all the text between the two points to appear in reverse field. Now you can select either Font/*/* or Style/*, and affect all of the text within the highlighted range. Unfortunately, you can only select one command when you set a range in this manner, so if you want to select a font and one or more enhancements, you will have to set the range for each command.

If you have purchased the Berkeley Softworks FontPak1, you may use these fonts to replace the original fonts on the disk. GeoWrite is limited to using any of the first eight fonts that appear on the disk, so you will be limited to eight fonts within any one document.

MARGINS

GeoWrite does not have many features for controlling the format of the text. Margins may be changed from one page to the next, tabs may be set, and spaces, carriage returns, and page breaks may be set. All measurements for margins assume that you are using an 80 dot per inch (dpi) printer.

The left margin marker appears as the normal field character "m" on the bottom part of the ruler line at the top of the geoWrite document window. The left margin may be moved from the extreme left of the document window (1.2 inches from the extreme left of the printed document page) to the 3 inch marker in the document window (3 inches from the extreme left of the document page). The minimum length of a text line on a geoWrite document page is 1 inch. That is, if the right margin is set to the left of the 4 inch marker on the ruler line, then the left margin marker cannot be set closer than one inch to the left of the right margin.

The right margin marker appears as a reverse field character "m" on the bottom part of the ruler line at the top of the geoWrite document window. The right margin may be moved from the extreme right of the document window (7.2 inches from the extreme left of the document page) to the 2.2 inch marker on the document window (2.2 inches from the extreme left of the document page), provided that the right margin is no closer than one inch to the right of the left margin.

The margins may be set by placing the pointer on the margin indicator, clicking the fire button once, and moving the pointer to the desired position. If there is a conflict of margins, geoWrite will set the margin indicator you are attempting to move to a position one inch away from the other margin indicator toward the moving margin indicator's default side. For example, if you have the left margin set at the 2.5 inch marker on the ruler and grab (select) the right margin indicator and attempt to drop it anywhere to the left of the 3.5 inch marker on the ruler, geoWrite will automatically place the right margin at the 3.5 inch marker on the ruler.

Margins may be set independently for each document page, and the margin settings for each document page will be maintained when the document is closed. Margin settings may also be restored by selecting the File/recover option from the command menu.

Margins also determine the position of photo scraps appearing on a geoWrite document page. Photo Scraps will be approximately centered between the left and right margin of the document page. If the Photo Scrap is longer than the distance between the left and right margins, the part of the Photo Scrap that would extend beyond the right margin is truncated. Any truncated portion of a Photo Scrap appearing in a geoWrite document is maintained in the file on disk and may be recovered be moving the margins. Truncated portions of Photo Scraps appearing in a geoWrite document will not be printed unless the margins are changed beyond the length of the Photo Scrap.

TABS

GeoWrite provides up to eight independent dynamic tabs. All tabs are optional and may be set anywhere between the left and right margin indicators. Tabs are useful for indenting the first line of paragraphs and for aligning columnar data. GeoWrite does not offer a paragraph indent function, nor can it be simulated by resetting the margins on the next page as geoWrite forces full page prints.

Tabs may be set anywhere on the ruler line between the left and right margin indicators. A tab indicator appears on the bottom part of the ruler as a small arrow pointing up toward the ruler line. Tabs may be set independently from one page to the next.

To set a tab, simply place the arrow pointer anywhere on the lower part of the ruler line and click. This will place a tab marker at the nearest one-tenth inch marker on the ruler line. You may move a tab by placing the pointer on the tab marker, clicking, and moving the pointer to the desired location. Tab markers may be placed only at the one-tenth inch markers on the ruler line. You may delete tabs markers by placing the pointer on the tab marker you wish to remove, clicking, dragging the tab pointer to the top part of the ruler line, and clicking again.

As opposed to the way many Commodore word processors work, the tabs in geoWrite are dynamic. That is, when one enters a tab into a document, geoWrite imbeds into the text an invisible control character [specifically, CHR$(9)]. This character is then present whether or not any tab indicators are set on the ruler line. If no tab indicators are set on the ruler line, the tab character can be present in the file and have no effect. Such invisible tab characters will affect the text as soon as tab indicators are set on the ruler line. Tab characters can be deleted from within the text by placing the cursor bar directly in front of the indented text and using the <DELETE> key, ^h, or the left/right cursor key.

Further, tabs may be placed after text on a line. That is, you may type in some text, enter the tab character (^i), and enter more text. The text on the line following the tab character will move to the right of the next set tab marker. If no tab marker is set after the position of the tab character on the line, then the tab character will be present in the text but will have no effect.

PAGES

By design, geoWrite handles all documents in terms of pages. Any time that a GEOS file is sent to the printer, a full page will be printed. Within geoWrite, text documents start at page one and, when the space occupied by the text exceeds the space available on one page, continues on to page two. When you cause geoWrite to move to another page, the page in memory will be stored to disk, and the data for the selected page will be loaded from the disk into the computer's memory. Only one page may be in the computer at any given time.

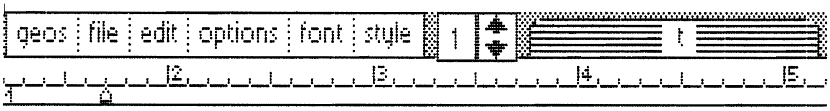
A page is limited by the computer's memory to no more than about 5000 characters. Attempts to create a page with more than 5000 characters will cause geoWrite to go into an endless loop as it tries to update the file on disk. If this occurs, you must turn off the computer and reboot GEOS. The file can be recovered by immediately going into geoWrite and loading the file that has more than 5000 characters on the document page and deleting some of the excess characters. When counting characters, it is necessary to count invisible characters such as hard carriage returns (one character), tabs (one character for each tab), and change of font/style flags (four characters) as well.

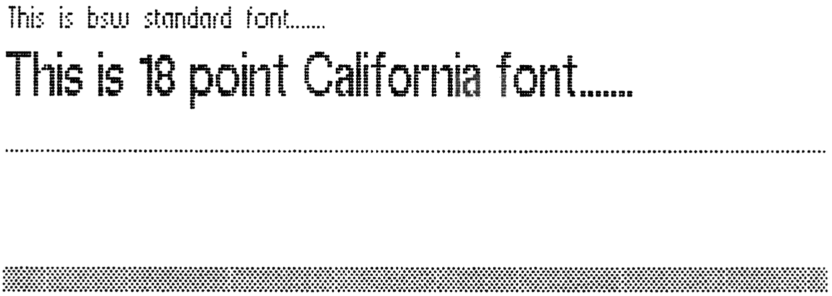
GeoWrite forces an end of page when the vertical length of the text data exceeds the length of a printed document page (about 10 inches, varying slightly with the combination of fonts and point sizes). This end of page is forced, and you must work around it. There is no way that you may change the spacing between lines (leading) without major software modifications. The end of page marker appears on the screen as a relatively wide band across the width of the document page.

You may force geoWrite to advance to the next page by selecting Options/page break command. Selecting this command imbeds an ASCII Form Feed [CHR$(12)] character into the text. This character is used as a flag to indicate that any text following the page break will be printed on the next document page. A page break character may be inserted anywhere on the document page, even in the first position to cause geoWrite to print a complete blank page. This page break character appears as series of evenly spaced dots across the width of the document page.

If a geoWrite document page is vertically filled (if you have reached the end of page), and you continue to type at the end of the page, geoWrite will force any text that you type onto the next page. If you are entering text on a filled geoWrite document page, but are somewhere on the page prior to the last line, any line feeds caused by word wrap, hard carriage returns, or the insertion of a Photo Scrap will cause the text at the end of the current page to cascade onto the following page. If the following page is also vertically filled, then the text at then end of that page will also cascade, etc.

If you are entering text onto a geoWrite document page that is not vertically filled, but terminates with a page break character, then the text at the end of the page will not wrap around until the document page is vertically filled, at which time the text, and the page break, will cascade onto the next page.




This is bsw standard font.......

This is 18 point California font......

HINTS & TIPS ON GeoWrite

As noted before, one the most notable handicaps of geoWrite is its speed. It is easy for even a hunt-and-peck typist to lose characters by typing too fast, especially when the full width of the page is used. The problems with speed cannot be wholly avoided, but you can work around them with a bit of care.

Much time is used by geoWrite in updating the screen. Any time that the document window moves against the background of the document page, geoWrite must move the data within memory and then redraw every character within the window. This can take a LOT of time. As an example, open a new geoWrite document, select the University 6 point font, and enter the letter "i." Keep entering it. And keep entering it. After a while, you will notice that geoWrite takes longer and longer to update the screen each time it has to move the window against the document page. This long time lag can be at least partially avoided.

To avoid this problem, consider first that geoWrite always fills the space between the margins. That is, as long as you do not enter hard carriage returns at the end of each line, geoWrite will automatically reformat the text when you change margins. It will even pull up text from following pages in order to fill the space. This makes it possible, and desirable, to reset the margins for entering text.

So, when you first create a document, move the pointer over to the right edge of the screen and so move to the right two-thirds of the document page. Then grab the right margin marker that sits on the bottom part of the ruler line, drag it over to about the five inch mark, and release it. This will move the right margin over. Now move back to the left part of the document page and enter your text. Resetting the margins in this way will prevent geoWrite from moving the window back and forth against the document page and save you quite a good bit of time.

GeoWrite also takes a good bit of time moving from page to page. Because GEOS uses so much of the computers memory for the operating system, only about five thousand characters can be in the computer at any one time. To allow for larger documents, only one page is stored in memory, and all the other pages are stored on disk. Then, every time you change pages, geoWrite saves the current page to the disk and loads the next, or previous, page from the disk. It must then redraw the page, etc., etc.

This will not be much of a problem for small documents. For large documents, however, the saving and loading of pages can take a considerable amount of time, especially if you try to insert text into a vertically filled page. Say you have five pages full of text, and want to insert a paragraph into the middle of the first page. Inserting the text will usually be no big problem, but as soon as you update the document, close the file, or move to another page, geoWrite must rearrange each and every one of those five pages. The text that you pushed off the first page must be put at the beginning of the second page, which pushes the end of the second page down to the beginning of the third page, etc. And each page is stored separately on disk. This can take more than just a little while.

To get around this problem, do not wait until a page is vertically filled and the end of the page is forced, but rather insert a page break. For example, say you are planning to write a ten page paper. The flow of the text will be fairly continuous, so you don't need to separate the pages with headers, starting a new topic or sub-topic at the top of a new page. Every time you get about two-thirds of the way down a page, select Options/page break. Do not worry about the extra space at the end of each page as you can take out the page breaks later. The important thing now is to prevent the slowness of geoWrite from getting your way as you put your thoughts on paper.

The advantage of this is that as you revise your text, geoWrite will not have to rearrange all the pages each time you insert or delete a line. Rather, as you insert a paragraph, it will force the page break that you inserted closer to the end of the actual end of the page. GeoWrite will not have to rearrange the pages until the length of the text before the page break character exceeds the length of the document page.

Similarly, do not let all the fancy fonts and enhancements get in the way of putting your ideas on paper. Get the ideas down first, then go back and change the fonts and enhancements.

This has two advantages. First, it saves time. The first time you load each font, geoWrite must load the font from the disk. If you use several fonts, geoWrite must often rearrange the fonts in memory, or even dump one font to load another. So changing fonts can take a good bit of time. The second advantage is that it helps discourage you from getting font happy. When you have so many fonts at your fingertips, it can be easy to try to use too many of them. Using too many fonts and enhancements can make the page noisy and detract from the text itself (although this can, at times, be a good idea).

In sum, as you enter text, draw the right margin in so the screen does not slide back and forth on you all the time, use page breaks so you do not grow old waiting for geoWrite to rearrange all the pages, and use a single font in a single size as you enter the text.

If you are lucky enough to have two disk drives, geoWrite will be much easier, but still not ideal. It would be quite handy to have all the programs, utilities, and fonts on one disk; then to have all the document data on the other. Alas. You may work with a geoWrite document that is on a disk other than the disk from which you booted geoWrite, but that is about it.

Say, for example, that you have a disk with deskTop, geoWrite, the printer module you need, and all the fonts and deskTop accessories on the disk in device 8. You have a blank disk in device 9. This does you absolutely no good.

If, however, the disk in device 9 DOES have a file on it, you can make that file almost as big as you could want. GeoWrite can handle a document as big as 64 pages, and each page can have as many as 5000 characters on it. This gives a total of about 320,000 characters - about twice as many as will fit on a single-sided Commodore disk. This makes it more difficult to create a big document. And by the time you consider the space on the disk taken by geoWrite, the fonts, deskTop, and any of the desk accessories that you might want to use, you really do not have much space left for your text.

So if you have two disk drives, you might want to sometimes take the long way around. First create a file. It does not matter what is on it; it could be completely empty. Copy that one geoWrite file to an otherwise blank disk. Now place that blank disk in one drive, a work disk with geoWrite, fonts, etc., in the other drive, select geoWrite/open/disk, and you are on your way. You can now create a single document that can hold almost 170,000 characters or about 30,000 words.

TEXT SCRAPS

A text scrap is a part of a geoWrite document that you have copied into a special file on the disk. To create a text scrap, set a range as you would to change fonts or enhancements. Then instead of selecting Font/*/* or Style/*, select Edit/copy or Edit/cut.

Edit/copy will copy the range of text that you have set into a file called "text scrap" on disk. The text in the document will remain unchanged.

Edit/cut will take the range of text that you have set out of the document and place it in a file called "text scrap" on disk. The text can be recovered with Edit/paste/text or with File/restore.

GEOS only allows one "text scrap" file on disk. An Edit/copy or Edit/cut will replace any text scrap that may previously have been on the disk.

To use a text scrap, select Edit/paste/text. This will place the text scrap into your document at the current position of the cursor. Generally, the text will need some little bit of reformatting as carriage returns in the text scraps, while predictable, are hard to follow.

If you wish to use more than one text scrap, please read the section on the Text Manager under deskTop Accessories.

PHOTO SCRAPS

A photo scrap is a part of a geoPaint document that you have copied into a special file on the disk. You can use this graphic by selecting Edit/paste/picture. The photo scrap will then be inserted into your geoWrite document following the current position of the cursor. If you insert a photo scrap into the middle of a text line, the line will be broken and the text following the cursor will appear below the graphic.

A photo scrap, once inserted, will be centered between the left and right margins. If there is not enough room between the margins for the photo scrap, the right edge of the photo scrap will be truncated.

If you need to use more than one graphic in a geoWrite document, and do not want to go back and forth to geoPaint all the time, read about the Photo Manager under deskTop Accessories.

PRINTING

All geoWrite documents may be printed either from within geoWrite or from the deskTop. To print from geoWrite, select File/print. To print from deskTop, you may either drag the document file icon to the printer or you may select the file icon and then select the File/print command.

geoPaint

GeoPaint is a nearly full featured graphics program that offers several features not offered by other graphics packages for Commodore computers. Most obvious, and the nicest, of these extra features is that geoPaint works with full pages as they will be printed on paper, rather than with files one or two screens large. Other nice features include the ability to copy portions of the screen to a special file, and to recall that portion of the screen elsewhere into the same or a different geoPaint document, or into a geoWrite document.

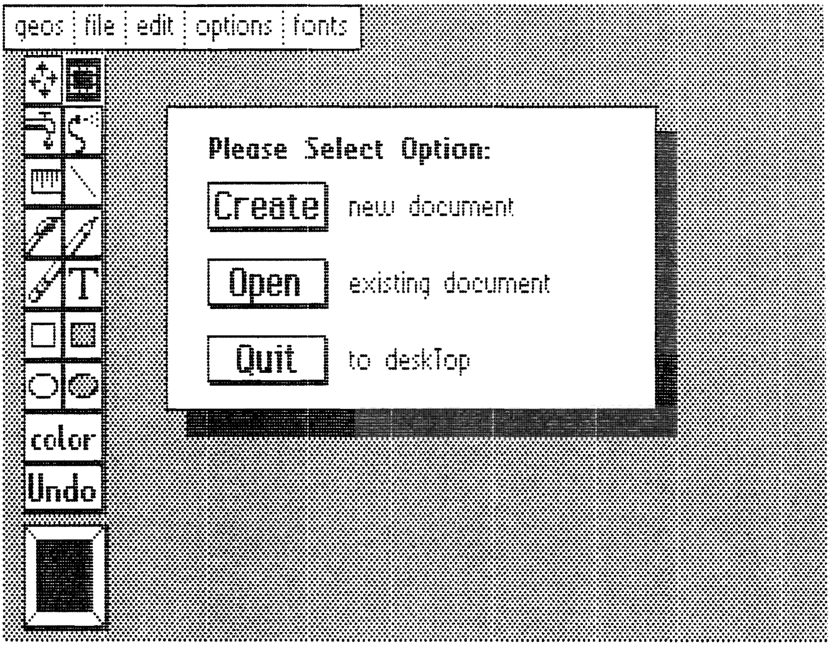
But, geoPaint is not perfect. It can be difficult to draw a straight line the length or the width of a page. It is very hard to draw a circle that is larger than one screen. And smooth curves such as an ellipse or parabola are almost impossible.

As graphics programs go, geoPaint is good, but probably not among the best. The features are similar to those found in many other packages, but some features are missing. Its real power lies in the ability to take graphics from geoPaint and put them in a geoWrite document. I know of only one other attempt to do this with a program for the Commodore 64, and the combination of PaperClip (tm) and Flexidraw (tm) is neither as powerful nor as easy to use.

From the GEOS deskTop, the first time you use geoPaint you must open the program file by either double clicking on the geoPaint icon, or by clicking once on the geoPaint icon and then selecting the File/open command. Later, after you have one or more documents already created on your GEOS work disk, you may run geoPaint from deskTop by selecting File/open or double clicking on any geoPaint document.

For example, you have used geoPaint to create a graphic document named "picture 1." You may enter geoPaint and edit "picture 1" by double clicking on the "picture 1" icon, or clicking once and selecting the 'open' command on the File sub-menu. Either way, you will enter geoPaint, and geoPaint will automatically load the "picture 1" document.

Any time you enter geoPaint by selecting the geoPaint icon from deskTop, the geoPaint application program will load and present you with the following screen.



Please Select Option:

Since this is the first time you are using geoPaint, choose the "Create new document" option. GeoPaint will then ask you to give a name for your new document, create a file on the disk with that name, and will then be ready for you to enter the text for your new graphic document.

If you want to edit a graphic document that you have worked on before, select the "Open existing document" option. This will present you with yet another menu. This menu gives you a list of geoPaint documents on the current disk. The menu window only shows the name of five files at a time, but you can click on the up and down arrows to show more files. This menu will not, however, show more that the first fifteen geoPaint files on the current disk. If you have more than fifteen geoPaint files on the disk, and you need to work with one that appears, say, in the seventeenth position, you will need to either rearrange the files on the deskTop Notepad, or you can open the seventeenth geoPaint file from deskTop.

If you are using two disk drives (devices 8 and 9), a third option will be presented on this menu. The "Disk" option will allow you to open an existing document that is stored on a disk in the other disk drive. Taking advantage of this feature will be explained more fully under Hints & Tips on GeoPaint.

When you first enter geoPaint, the program defaults to the Pencil drawing tool. If you wish to use any of the other drawing tools available, you need only use your joystick or mouse to place the pointer on the tool you want to use, and click.

USING GEOPAINT

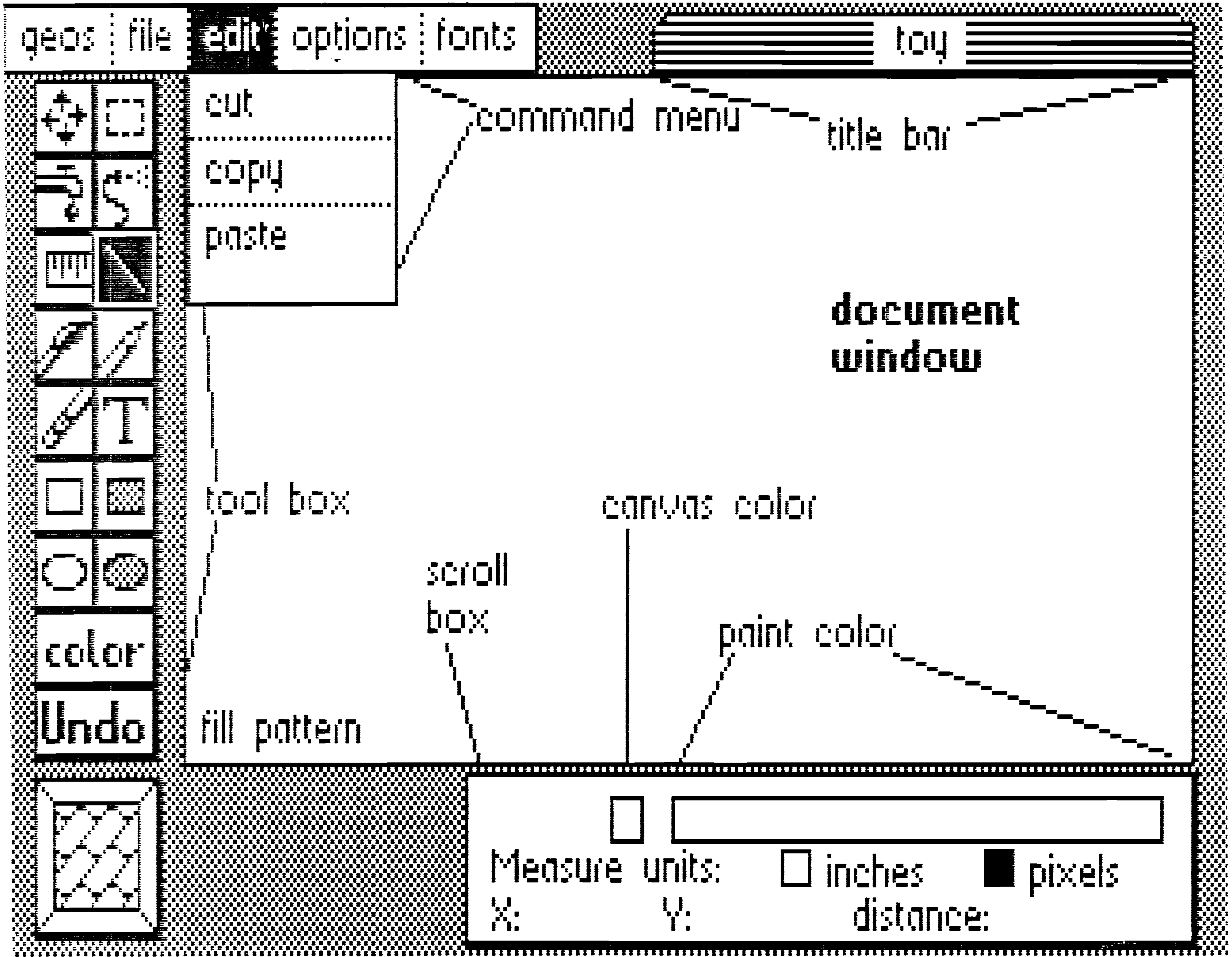
Layout & Features

The geoPaint screen bears some resemblance to the deskTop and geoWrite screens, namely the Title Bar and the Command Menu. Also, geoPaint has a main work area similar to the document window in geoWrite. GeoPaint, however, has a far greater number of tools; more options; and menus and sub-menus that are not offered elsewhere in GEOS.

First, of course, is the pointer. Unlike other GEOS applications and accessories, the pointer can do many things in geoPaint other than simply select and position. When the pointer is outside the geoPaint window, it acts just like the pointer in deskTop or geoWrite. Once you move it into the document window, however, it becomes a geoPaint tool. That is, if you click while the pointer is within the document window, the active tool will begin to function. When you first enter geoPaint, the default tool is the pencil. If you click the button while the pointer is in the document window and move the pointer, it will act like a pencil and draw a line as you move. Click again to deactivate the tool.

As you enter a geoPaint document, most of the screen is occupied by the window. This window shows a small part of the total geoPaint document page. The position of the window against the virtual document page is shown by the black rectangle against the gray box in the bottom center of the screen. This is called the Status Box in the GEOS User's Guide, but we will refer to it as the Scroll Box here.

GeoPaint provides two methods of moving the window against the document page. The easiest is to place the pointer in the Scroll Box and click. This will cause the pointer to grab the black rectangle (window icon) and turn it blue. You may then place the window icon anywhere on the page that you wish and click again to release it. GeoPaint will then save the data from the current screen to the disk, load the data for the other part of the screen from the disk, and then redraw the screen.



Measure units: □ inches ■ pixels
X: Y: distance:

At the top right corner of the screen is the Title Bar. Here geoPaint shows you the name of the graphic document that you are currently working on.

At the top left of the screen is the Command Menu. This Command Menu is very similar to the Command Menu of the deskTop screen, but has some different sub-menus and some different options within some of the sub-menus that remain.

At the extreme left is the GEOS sub-menu. The commands here provide information about geoPaint and allow you to use any of the first eight Desk Accessories on the currently selected disk. Options not available here, but available from deskTop, are "select printer," and "select input."

The FILE menu provides several options different from those offered in deskTop.

close: selecting this option will save the currently loaded document to the disk, and return you to the first geoWrite menu (Create/Open/Quit);

update: selecting this option will save the currently loaded document to the disk and return you to the document. This option is useful when you are working with a particularly long document, have made several changes to the document, or are about to try something with the document and are not sure how it will turn out;

preview: selecting this option will cause geoWrite to show you a graphic depiction of the complete document page as it will appear when printed. The Commodore 64 does not have sufficient resolution to produce such small text legibly, so the page shown by the preview option will be unreadable. It will, however, give you a good idea of how the text will appear, prior to printing. This option will show only the current page. To preview a multi-page document, you must move from one page to the next, and select the File/preview option for each page;

recover: selecting this option will, in most cases, allow you to recover from the changes you have made to the current page since the last time you saved the current page. Recover will NOT work if you have moved from one page to another. This option is particularly useful if you have cut a large piece of text from the page, and wish to have it back. If you attempt to recover, and a window opens telling you that geoWrite "Cannot recover from file," try again. Many times, the recover option will work on the second attempt. The recover option, however, is not wholly reliable. If the File/recover option does not work, you might want to try the Edit/paste/text option to recover lost text. The File/recover option can also be used to remove additions made since the last time the document page was saved;

rename: selecting this option will cause geoWrite to prompt you for a new name for the currently loaded document, and then to rename the document in the computer and on disk;

print: selecting this option will cause geoWrite to translate the currently loaded document, using the selected printer driver's translation table, and send the interpreted data to the printer;

quit: selecting this option will cause GEOS to save the currently loaded document as is to the disk from which the file was loaded. If the deskTop program file is on the disk, then GEOS will return to the deskTop. If deskTop is not on the disk, then GEOS will prompt for a disk that has deskTop on it. Sometimes, however, GEOS will go away and take your computer with it. If this should happen, the file will be intact on disk as it was in the computer when you selected the File/quit option, but you will have to reset your computer and reload GEOS.

Two of the remaining Command Menu options have more to do with tools than with standard commands. The Edit/* commands and the Font/(font)/(size) commands interact very deeply with tools and the Status Box. Therefore, explanations of these commands will be deferred until the appropriate tool is discussed.

The Options command, however, doesn't really fit anywhere. Two of the Option commands (Options/pixel edit and Options/normal edit) could have been combined into one toggled command; one is a toggle (Options/color), and the other (Options/change brush) could have been handled in the Status Box when the Paint Brush tool is selected. But, these are in the command menu, and so are explained here.

pixel edit: this command allows you to edit data in the current window on a much finer scale than the normal edit mode. When this command is selected, a blue-cornered frame will appear in the upper left of the window. You may move this frame around the window using a joystick or mouse. Once the frame surrounds the region you wish to edit, click the fire button. GeoPaint will expand that region to fill the full window, and will show the same area in standard size at the bottom of the screen to the left of the status box.

The pixel edit window represents 65 pixels across by 35 pixels down, or eight-tenths of an inch by just over four-tenths of an inch.

Several tools are available within the Pixel Edit window, namely Fill, Airbrush, Ruler, Line, Paint Brush, Pencil, Eraser, Open Box, and Filled Box. The Undo command in the Tool Box will function as usual. Selecting the Scrolling Arrows (at the top of the left column of the Tool Box) will return you to the normal scale window and retain the blue-cornered frame, allowing you to move to another region to edit by pixel.

When working within Pixel Edit, not all changes will be immediately apparent on the large window, but all changes will be visible in the smaller window to the left of the Status Box.

normal edit: selecting this command will take you out of the Pixel Edit mode.

change brush: this command provides options for controlling the size and shape of the brush that will be used when the Paint Brush tool is active. Selecting this command will cause geoPaint to replace the Status box with the brush menu which contains thirty-two different brushes. To select a brush, move the pointer to the desired brush and click. The Paint Brush is affected by the fill pattern - the brush will paint into the document the current fill pattern, masked by the selected brush. For example, if you select the fourth brush from the left on the bottom row of the brush menu, and select the brick fill pattern, when you activate the tool in the drawing window and move it across the page you will get the brick pattern the size of the brush. This provides for a great variety of combinations with some interesting effects.

color off/on: (toggle) this command will turn the color on or off. When color is toggled off, the color bar in the Status Box will be deactivated and the Color command in the Tool Box will be disabled. Although color pictures are often more attractive, the color can be a hindrance when you print the graphics on a single color printer. Commodore graphics treat all colors as either on or off. Selecting the Options/color off command will show you how the document will appear when printed. Also, graphics used in geoWrite will not be printed in color, even on a color printer. Thus, this option will show how a graphic will appear when it is moved through the Photo Scrap to geoWrite.

The remaining options in the Command Menu - Edit, Options, and Fonts - all interact with tools selected from the Tool Box. At this point, the selection of geoPaint options can become quite complex. For example, to use the commands offered in by Edit in the command menu, you must first select a region of the current window by using the Edit tool or the Text tool. Selecting either of these tools will also change the information and options given in the Status Box in the lower right of the screen. The remaining Command Menu options will be explained here with reference to the appropriate tools and changes to the Status Box. More information will be given in the explanation of each tool.

The EDIT menu provides options for cutting, copying, and pasting data from the current geoPaint document. To use these commands, the Edit tool must first be selected from the top of the right column of the Tool Box, and used to set a region within the geoPaint window. Alternately, you may set a range using the Text tool for cutting, copying, and pasting text data. The Edit commands will have no effect if you have not set a range on the geoPaint screen. Once you have selected an area, you may use the Edit command to:

cut: selecting this option will cause geoPaint to delete the area of graphics set with the Edit tool, or the range of text set with the Text tool, from the current document, and to place the deleted graphics into the appropriate Scrap file on disk, thus replacing any material that may have previously been placed in the Scrap on disk by the Edit/cut or the Edit/copy commands. If the Edit/cut option is selected with no region of graphics selected, the Photo Scrap file on disk will not be replaced and the document will not be affected;

copy: selecting this option will cause geoPaint to copy the selected region of graphics from the current document into the Photo Scrap file on disk. The document itself will not be affected. If the Edit/copy option is selected with no range of graphics selected, the Photo

Scrap file on disk will not be affected. Graphics copied with the Edit/copy option will replace the contents of the Photo Scrap file on disk;

paste: selecting this command with a range set by the Edit tool will cause geoPaint to copy into the selected region the graphic data that is stored in the Photo Scrap on disk. Selecting this command with a range set by the Text tool will cause geoPaint to copy into the selected region the text data that is stored in the Text Scrap on disk. If the selected region is smaller than the original size of the Photo Scrap, only that portion which will fit into the selected region will be copied into the current document.

Once a Scrap has been pasted into the selected range of the current document, it may be moved by placing the cross-hair within the region. Then move the cross hair to the desired location within the window and click again to drop the region. This move function will not allow you to force the region across the border of the window.

Be careful! Pasting a Scrap will overwrite all data within the region set with the Edit or Text tool! Moving the Scrap will overwrite all data within the region where the Scrap is dropped. Further, the Undo command in the tool box is only partially functional. Undo will not remove the Scrap data once the Edit/paste command has been selected. It will, however, reverse the last Move action. Once you have selected Undo from the Tool Box, you will not be able to move the region. And, if you select Undo, the dotted lines that define the Edit region will be stored into your current document.

Edit/paste is a very powerful command, but use it with caution. If all else fails, you may use the File/recover option to restore the current window to how it was since your last window move.

TOOLS

GeoPaint provides a wide variety of tools for creating a graphic document. Most of these tools are used to enter the actual graphics data, but a few of the tools provide options for moving around the document, working with Scraps, or otherwise controlling the document. Many of the tools change the options within the Status Box or interact with the options in the Command Menu. Most of the options of the Status Box will be documented here; those not discussed here will be explained in the next chapter.

The Tool Box is arranged in two columns at the left side of the screen. To select a tool, move the pointer to the Tool Box and click. Many tools are stored on disk and so will require a quick load from the drive. Most tools will have no effect until the pointer is moved into the document window and the fire button is clicked again. We will discuss the tools in sequential order starting at the top, from left to right.

scrolling arrows: selecting this tool will update the current document file on disk, and then place a picture of the tool in the middle of the screen. You may then move the window around the document page incrementally by pushing the joystick or mouse in the desired direction. Pushing up moves the window up relative to the virtual page. File/recover will not effect anything once the Scrolling Arrows have been selected.

edit box: this is a very powerful tool. By itself, the edit tool simply sets a range of graphic data for manipulation by the options provided by Edit/* in the Command menu and other commands provided in the Status Box. Once the Edit Box is selected and the pointer is moved to the drawing window, the Edit Box tool will appear as cross-hairs. Clicking once will set one corner of a region, and a second click will set the opposite corner. A region set in this way may be as large as the drawing window.

Once a region has been set, you may use the Edit/* command to cut, copy, or paste graphics data into the selected region.

edit/cut: this command will copy the graphics data from the selected region into the Photo Scrap file on disk. Any data previously in the Photo Scrap file will be replaced and the graphics data from the selected region will be removed from the current document. If you choose Edit/cut by mistake, you may recover by using Undo, or Edit/paste. The Photo Scrap file on disk, however, cannot be recovered from within geoPaint.

edit/copy: this command will copy the graphics data from the selected region into the Photo Scrap file on disk. Any data previously in the Photo Scrap file will be replaced. The current document will be unaffected.

edit/paste: this command will copy the graphics data from the Photo Scrap file on disk to the region set by the Edit tool. If the Scrap is larger than the region, only that part of the Scrap that fits will be inserted into the current document. If the region is larger than the scrap, the complete scrap will be copied into the current document. In either case, the top left of the Scrap will be put at the top left of the region set. Using the Edit/paste command will overwrite all the data behind the Scrap. If the Scrap is larger than the set region, then the whole region will be overwritten. If the region is larger than the Scrap, then the area the size of the scrap, starting in the upper left corner of the region, will be overwritten. The Undo command from the Tool Box will not work. To recover from an Edit/ paste, you must use the File/recover command. This will return the current window as it was the last time you moved the window or selected File/update.

The Status Box offers still more options for manipulating the data within the region set by the Edit tool:

move: The Edit tool defaults to this function. To use this option, you must first set a region using the Edit tool. Then place the cross-hair pointer inside the region you have selected and click. This will allow you to drag the set region anywhere within the current window. Pausing for a moment will allow geoPaint to redraw the current window, showing how it will appear if the region is moved. Click again to drop the region. The complete set region will overwrite the area on which it is set. Errors can be corrected with the Undo command from the Tool Box. Move will erase the region from its origin.

copy: functions similarly to Move, above, except that it does not erase the region from its origin.

mirror x: will transpose the set region, placing the left edge of the region at the right edge, the right edge at the left. It actually works as if the region had an axis running vertically through its center and you spun the region one-half of a complete turn. You may recover from a Mirror X with the Undo command or by immediately selecting Mirror X again.

mirror y: similar to Mirror X, transposing top to bottom, bottom to top.

rotate: selecting this option will rotate the data within the selected ninety degrees in a clockwise direction. Rotate may be selected multiple times to effect a rotation of one hundred and eighty or two hundred and seventy degrees. Rotating is a tricky, almost dangerous function. For example, on an empty window draw a filled box that occupies most of the window. Now use the Edit tool to set the region slightly larger than your full box. Rotate. Now rotate again. The resulting box will only be a part of your original box. Further, when the region is rotated, all the data behind the rotated region will be replaced. Thus you can unintentionally lose some graphic data and you must use File/recover to undo any damage. Be careful!

invert: selecting this option will reverse the colors within the selected region. This option, too, can produce some interesting effects, especially when you change colors often. Invert can be used to recover from itself if used again immediately.

clear: will erase all graphic data from the set region, but not the color data.

faucet: using this tool will cause geoPaint to fill an area with the currently selected fill pattern. The area filled will be the whole window unless an area is completely enclosed. For example, draw a relatively small hollow circle in the middle of the screen and then use pixel edit to break the circle by one pixel. Return to normal edit, select the Faucet tool, and click inside your now broken circle. The entire screen will fill with the current fill pattern. If this happens, you will have to wait until the fill is completed, then select Undo or File/recover. The Fill Pattern is explained more fully below.

Some very strange effects can be accomplished by using the Faucet with a number of various patterns repeatedly. The Faucet works by checking whether a particular pixel is ON or OFF. It will change all connected pixels from OFF to ON (or, if all the pixels are ON, it will change them to OFF) according to the selected fill pattern. If you have two black boxes on the screen connected by a line one pixel wide, the

Faucet will usually fill both boxes with the selected pattern. Fill patterns are always properly aligned with themselves. For example, if you use the Faucet to fill a region and then use the Paint Brush within the same region and the same fill pattern, you will not notice any effect.

Another interesting feature of the Faucet: generally, the Faucet will be used in an area that is constant, but it does not have to work this way. For example, whether you have an hollow or a solid filled circle, you can use the Faucet to fill it with a pattern. But if the circle is already filled with a pattern, you can still use the Faucet to get some interesting effects by placing the cross-hair pointer on part of the fill pattern that is on and filling the fill pattern. The fill pattern will follow the dots that are on, ignoring the empty space between them.

airbrush: this tool will spray certain parts of the selected fill pattern. The actual method of doing this gets quite complicated. The Airbrush has a certain pattern of dots. If the Airbrush tool is active, and the dots of the Airbrush match the dots of the fill pattern at the particular location of the Airbrush, then the dots at that location will be reversed (turned ON if they were OFF, OFF if they were on). Thus the Airbrush is not even pseudo-random. For example, select one of the brick fill patterns using the Pattern Fill. Then use the airbrush in any portion of the screen, going over the same area again and again. The area you are going over will eventually show the selected fill pattern.

ruler: this tool can be used to measure any distance on the current window in either pixels or in inches. Using the Ruler will make no changes to the document. Measurements in inches assume that you are using an 80 dot per inch printer. To use the Ruler, first select the tool and move the pointer into the window. Place the cross-hair pointer at one end of what you wish to measure, click once, and move to the other end.

Clicking a second time will clear the Ruler and allow you to move the pointer to another starting point. The Status Box at the bottom will show the difference between the first and second points - the difference horizontally (x), vertically (y), and total distance between the two points. The Ruler defaults to measuring in pixels; if you wish-to measure in inches, you must select inches from the Status Box before setting the first point for your measurement.

line: this tool allows you to draw a nearly straight line as possible within the current window. After selecting the Line tool, move the pointer into the window and set the first point by clicking. Move the cross-hair pointer to other end, and click. The resulting line will be as straight as the C64 allows. The last line drawn can be erased with the Undo command.

paintbrush: this tool is similar to the Airbrush tool except that it allows you to choose the pattern of dots that will be used to compare against the fill pattern, where the Airbrush has the fixed pattern. The Paintbrush will, depending on the brush selected, either completely replace with the selected fill pattern the area you paint, or change bits in the same manner as the Airbrush. For example, if you use the default Paintbrush (a small square) and the brick fill pattern, the Paintbrush tool will cover with the brick pattern any area over which you move the Paintbrush. If, however, you use one of the Paintbrushes found to the lower right of the status box, the Paintbrush will only change those dots that the Paintbrush actually hits.

To change Paintbrushes, select Options/change brush from the Command Menu. This will replace the information in the Status Box with a menu of brush sizes and types. Place the pointer on the brush you wish to use and click.

pencil: default. This tool appears as a small pointer arrow within the drawing window. Clicking once will activate the Pencil, allowing you to draw free-hand. Clicking a second time will deactivate the Pencil. Undo will remove the lines drawn the since the last time the Pencil was activated, provided no other commands or tools have been selected.

eraser: this tool will allow you to blank areas of the screen to the background color. It appears as a small square within the window when selected. The Undo command will work as normal.

text: this is a very powerful tool, similar to the Edit Box tool in the way it works, but designed to control text rather than graphic data. The Text tool allows you to set a region for the entry and manipulation of text, and interacts with the Edit/* commands for storing and recovering Text Scraps on disk, and the Status Box for enhancing text.

When selected, the Text tool will appear in the window as cross-hairs. You may then set a region in the window in the same manner as you set an Edit region. Move the cross-hair pointer to the one corner of the region you wish to set, click once, move to the opposite corner of the region, and click again. Unlike the Edit region, however, you may move a Text region once it has been set. For example, use the Text tool to select a region. Once you have set the second corner of the region, a flashing bar cursor will appear in the top left corner of the region. Enter a few characters, then click again. It will almost appear as if the Text region, and the text within it, has disappeared. What you have actually done, however, is set the first point of another text region. If you move the cross-hair pointer again, and click, you will have another text region with the same text inside it, again at the upper left corner. The text region will only overwrite the graphics behind it when the characters impose upon the graphics. A more complete explanation of this is given below.

Once you have set a region, you may select fonts from the Font/(font)/(size) command. Any of the first eight fonts on the current disk may be used, but only one font may be used within any one text region. Likewise, you may select various enhancements and color from the Status Box at the bottom right of the screen. Any enhancements will affect all text within the current region.

Having set a text region, the Edit/* command may be used as follows:

edit/cut: this command will copy the text data from the selected region into the Text Scrap file on the current disk. Any data previously in the Text Scrap file will be replaced, and the text in the selected region will be removed from the current geoPaint document. If you choose Edit/cut by mistake, you may recover the data in the document by using Undo or Edit/paste. The previous Text Scrap, however, cannot be recovered from within geoPaint.

edit/copy: this command will copy the text data from the selected region into the Text Scrap file on disk. Any data previously in the Text Scrap file will be replaced. The current document will be unaffected.

edit/paste: this command will copy the text from the Text Scrap file on disk into the currently set region of the geoPaint document. This can get rather complex. See the comments below on reformatting text in geoPaint.

style: the Status Box at the bottom of the screen gives options for color and for different combinations of enhancements. The selected color and enhancements will affect all text within the current text region. Excepting the plain text option, all enhancements are toggled. That is, if you select italics then decide that you don't like it, you can select italics again to put all the text back upright. You may use any combination of these options. Selecting plain text will return all text in the text region to the standard unenhanced font.

Reformatting Text in geoPaint

The way geoPaint handles text is really rather marvelous. Let us start with an example. Use the Text tool to set a text region across the top of the window. Now enter enough characters to span the width of the window to and begin a second line. You may now place the I-beam cursor anywhere within the active text to move the flashing bar cursor. Editing text works very similarly to geoWrite, minus amenities such as tabs.

Now that you have some text entered, move the I-beam cursor outside of the text region, and click. Although it seems that the text has disappeared, it is still there. What you have done is set a new first corner for a text region. Move the cross-hair pointer somewhere across the screen, wait until the region is stable, and click again. You now have, again, the text within the region. If the new region is not as wide as the original region, the text will be formatted to the new, narrower width. Words will remain intact as much as possible. In this way, the text region and the text within it may be moved about the window. This works also with text pasted into the region using Edit/paste.

When you enter text into a geoPaint document, graphics data behind the characters will be overwritten, but the other graphic data within the region will remain untouched. You may freely move the text around in this manner without fear of erasing the graphics behind the text. The graphic data will be permanently overwritten only when you select another tool or command. If you enter text into a text region and then select Pencil, for example, the Undo command will not work, and you must use the File/recover command to correct any mistakes. You may, however, use Undo if you have selected the Text tool again.

Only one font and combination of enhancements may be used with any one text region. This is not to say, however, that you are limited to one font and combination of styles for each document, or even each window. Page 63 shows an example of Roma 12 point plain text and Roma 12 point bold. This was done using text regions repeatedly within the same window. First set one text region, enter your text, and move it to where you want it on the window. Now move the pointer back to the Tool Box and select the Text tool again. Now you can set another text region and use another font and combination of enhancements within this region. You can do this as many times as you wish.

Be careful, though. The text regions can be moved, and will reformat the text when you move, only until you select a command or tool. Once you have selected a command or tool - even the Text tool - the current text region is translated from text data to graphic data. It can no longer be moved into the Text Scrap with Edit/copy or Edit/cut commands. It can, however, be moved, copied, or erased using the Edit tool, but you lose the powerful text reformatting of the Text tool.

hollow box: this is a simple tool that allows you to draw the outline of a box. This tool will appear as cross-hair pointer in the window. Click once to set one corner of the box, click again to set the other corner of the box. The box may be made smaller or larger until you have selected the second corner of the box. The status menu will show the length of the horizontal line of the box (x); the length of the vertical line of the box (y); and the distance from one corner to another. The measure defaults to pixels. To measure in inches, you must select inches before you set the first point of the box. Keep watch on this, as what appears as a square on the screen will not print as a square on paper. For a square to appear as a square on paper, the box must have the same X and Y lengths. (This assumes that you are using an 80 dot per inch printer. If you are using a 60 dot per inch printer, you will need 60 x pixels to every 80 y pixels for a square to come out square on paper.)

filled box: this works the same as Hollow Box, except the box will be filled with the current Fill Pattern. This will overwrite any and all graphic data behind the box that you draw. Each fill pattern is consistent to itself across the document, so you can draw one Filled Box with a pattern, then draw another Filled Box with the same pattern and if the boxes touch, the patterns will align.

hollow circle: similar to hollow box, except the Status Box does not show the size.

filled circle: similar to Filled Box, except the Status Box does not show the size.

color: this tool allows you to change the canvas and paint color of one or more cards on the current window. A more complete discussion of color in geoPaint is given below.

undo: this command will allow you to recover from many actions. It does not, however, work for all options or fully recover from some others. It will not, for example, completely recover from multiple uses of the rotate option of a graphic edit region. Undo is designed to recover one step back. That is, if you draw two lines with the Pencil, selecting Undo will erase the second line. Selecting Undo again will neither erase the first line nor undo the Undo and replace the second line. The Undo command will often not work properly, if at all, with the Edit and Text tools.

The Current Patterns Indicator box at the lower left is fairly well explained by its name. The pattern contained within this frame will be used by several geoPaint tools either directly or indirectly.

Clicking with the pointer on the Current Pattern Indicator will cause geoPaint to replace the Status Box with the Patterns options. Once this has been done, you may not return to the drawing window until you have selected a pattern.

It might be best to think of the fill patterns as always being present on the screen, but hidden. Each fill pattern is consistent with itself across the whole document. If you draw a small Filled Box using the bricks pattern in the upper left of the document page, and a similar box in the lower right of the document page, and another box in the middle of the document page, print it, and then compare it to the printout of a document page that is nothing but the brick pattern, all the bricks will be in the same place. This is true of all the patterns. The only way to offset a pattern from its normal position is to use the Edit tool and move, copy, or send it through a Photo Scrap.

When a pattern is used in conjunction with a tool, the pattern may work either cover all the area affected by the tool or work in conjunction with the tool as a mask. The Filled Box and solid Paint Brushes are good examples of the former. When these are used, the pattern will entirely cover the area of the Filled Box or the area over which the Paint Brush is drawn.

The latter, when the fill pattern is used as a mask, is more interesting. In this case, you really have two patterns working together. One is the pattern as shown in the Current Fill Pattern Indicator. The other pattern is the tool you are using. The Airbrush and non-solid Paintbrushes are examples of this. With the Airbrush, the pattern of the Airbrush, as shown by the dots of the tool when it is within the window, is periodically compared to the Current Pattern. The dots that match when this comparison is made are then written on the document. Non-solid Paintbrushes work the same way except the comparison is made constantly while the Paintbrush tool is active.

The effect of this masking is that you can get different effects by selecting different Current Fill Patterns with different brushes. A non-solid Paintbrush will give continuous lines when used with the solid fill pattern, but will give very discontinuous lines when used with, say, the slanted line fill pattern.

One special effect that is particularly easy to create is a foggy or very aged appearance. Select the brick fill pattern and use the Airbrush tool. Activate the Airbrush within the window, draw it back and forth horizontally several times across the same section of the screen, and use your imagination. It looks somewhat like a brick wall seen through thick fog. You can make it more solid by continuing to draw the Airbrush over the same area of the screen.

STATUS BOX

The Status Box that appears at the bottom right of the screen has several different functions depending on the tool being used. Most of these have been explained along with the appropriate tool, but we will go over them again here, starting with the one that has not been covered at all. The Status Box defaults to showing the position of the drawing window, and the current canvas and paint colors.

On the left side of the Status Box, a large grey rectangle represents the complete document page, and the smaller black box within the grey rectangle represents the drawing window. If you place the pointer within this Page Map, the little window will attach itself to the pointer, allowing you to place the window anywhere on the page you wish. Clicking again will release the little window from the pointer and cause geoPaint to update the document file on disk and load the data, if any, for the selected portion of the document. This is a relatively quick way to move a long distance across a document.

The little box set aside just to the right of the Page Map indicates the current background color. This will change as you use the Color command from the Tool Box, or as you move across different background colors within the current window.

Finally, you may select the color with which all the tools will draw by placing the pointer on the desired color in the Color Menu in the Status Box.

COLOR

GeoPaint is a high-resolution graphics editor. As such, the hardware limitations of the Commodore 64 limit the use of color within geoPaint. The Commodore 64 works with all graphics in terms of 8 pixel by 8 pixel cards which are in fixed positions on the screen. There are one-thousand of these cards, forty across and twenty five down.

The Commodore 64 allows four colors in each card in low resolution mode, or two colors in each card in high resolution mode.

Because geoPaint works in high resolution mode, you may only use two colors within each Commodore 64 graphics card. The foreground, or Paint, color for each card is shown in the default Status Box. The background, or Canvas, color defaults to the plain grey of the geoPaint drawing window.

The Paint color can be changed by selecting a color from the Color Bar in the Status Box. Once a color is used in a graphics card, the Paint color for that card is changed to the color used. If you draw a black solid filled box in the middle of the drawing window, and then draw a white line through the middle of the black solid filled box, the white line will be one pixel wide outside the box, then expand to a white bar one card wide as it goes through the box. If the white line is neither vertical nor horizontal, the black solid filled box will be broken by a staggered pattern of white cards.

This is because geoPaint can use only one canvas color and one paint color in each card. Each dot within a card is either

ON or OFF. If the dot is ON, it will show the paint color. If the dot is OFF, it will show the canvas color. As soon as you turn on one dot using any tool in any card, the paint color for that card will become the current color as shown in the status box. This can make it quite difficult to use several colors in a small area.

The only way to change the canvas color is to select the Color command from the Tool Box.

As you can see, color is difficult to work with in high resolution mode on the Commodore 64. Berkeley Softworks could have allowed the use of low resolution mode, which allows for four colors in each card. This is another of the trade-offs that must be made. GEOS is designed for creating documents for printing, and few users have color printers. Limiting geoPaint to high resolution mode allows most users to take advantage of the resolution of the Commodore 64 and the resolution of their printers.

Thus, unless you have a color printer, or you wish to create a geoPaint document to look at and don't care too much about printing, you may want to limit yourself to two colors for the whole geoPaint document. This is because, when GEOS prints a geoPaint document, it prints all the dots that are on, and leaves blank all the dots that are off. When working with a color document, it is easy to become confused and switch the canvas and paint colors. When printed, the dots may then be reversed. To preview how a document will appear when printed on a single color printer, use the File/preview command. To quickly view how a single window will be printed on a single color printer, select the Options/color off command.

Color will not be used when a Photo Scrap is used in a geoWrite document, nor will color appear when a Photo Scrap is shown in the Preference Manager. Color data will be maintained in the Photo Scrap and in the Photo Manager.

HINTS & TIPS ON GeoPaint

As noted before, one of the most severe handicaps of geoPaint is the way it handles color. This handicap is compensated for by the availability of higher resolution. Because of the limitations on color, it is often a good idea to turn the color off completely and work with light and dark gray only. This will more correctly show the geoPaint document as it will be printed on a single color printer.

Another shortcoming of geoPaint is that you can only work within the current window. This is especially troublesome when working with circles or trying to draw long straight lines. If the circle you are drawing extends beyond the limits of the current window, every part of the circle beyond the edges of the window will be lost. GeoPaint cannot draw a circle, square or line that extends beyond the current window.

There is no good, simple way to complete a circle larger than one screen window.

Straight lines and boxes larger than a screen window are relatively easy to create. Start the line or box in one screen window then use the Scrolling Arrows tool to move the window a short distance against the document page. This is a bit slower than it might be, but does allow for consistency. This will also work correctly with Fill Patterns, as the patterns are consistent to themselves, and will not skew from one region of the document page to another.

Another missing feature is a way to set a point. The lack of this feature makes it difficult to draw concentric circles or several lines all starting at the same point. Every time you move the pointer, you lose the set point, with the exception of with the Line tool. With the Line tool, you may set one end of the line, move the pointer, click to set the second end of the line, and click still again to set the first point of the second line. This is acceptable for drawing a triangle, but not sufficient for drawing several straight lines with one shared point such as the illustration on page 4-10 of the GEOS User's

Guide. The joystick does not have fine enough control to set a point within one pixel consistently.

This is complicated further by the use of the joystick and pointer to select all commands and options. For very precise work, it would sometimes be nice to set a point using pixel edit, and leave the pointer there as you return to normal edit. If this were allowed, you would be able to set single points with greater precision.

GeoPaint provides no absolute scale. The Scroll Box that is given at the left side of the Status Box gives you an approximation of your position on the geoPaint document page, but not absolutes. If the exact position of the pointer on the document page could be easily determined, geoPaint could be an almost respectable CAD program.

GeoPaint will occasionally get lost. For example, use the Line tool in normal edit, and then select Options/pixel edit. The Line tool will still be active, but the measure units in the Status Box will be lost. Still while in pixel edit, select the Line tool again and the measure units will be shown in the Status Box. The same will happen with the Box tools and even the Ruler. To use the measure units in the Status Box, the tool must be selected within pixel edit. If a tool is selected within pixel edit, however, the measure units will still be shown when you return to normal edit.

Finally, update your files on disk frequently, especially if you are working on a particularly complex graphic document. Update frequently so if you make a mistake you can recover all but the most recent changes, and update frequently so geoPaint does not corrupt your document. This will happen only rarely, and only after particularly long sessions, but it can. Keep an eye on all features of the screen. If garbage ever appears anywhere on the screen, quit to deskTop and reload geoPaint. If garbage appears on the screen, options may begin to work oddly or not at all, the pointer will randomly become an active tool, and you can corrupt your document.

PRINTING

A geoPaint document may be printed from within geoPaint by selecting the File/print command, or from deskTop by dragging the document icon to the printer.

GeoPaint assumes that geoPaint documents will be printed on an 80 dot per inch printer. All scaling assumes an 80 dot per inch printer. For example, if you draw a horizontal line 80 pixels long, GEOS will use 80 printer dots to print a horizontal line one inch long on paper.

If you are using a 60 dot per inch printer, GEOS will still print 80 dots. Thus the 80 pixel line drawn with geoPaint will still be printed as 80 printer dots, but the space between the printer dots will be farther apart, yielding a line one and one-third inches long.

Vertical length is similarly affected, although printers are not so neatly categorized. GeoPaint works on the assumption that printers will print 80 dots per vertical inch as well as 80 dots per horizontal inch, and provides 800 vertical dots on the document page. This allows for a document to be 10 inches long when printed. If your printer does not print vertically at 80 dots per inch, the vertical length of your document will vary, but usually within a smaller range.

If you are using a 60 dot per inch printer, your graphic documents will appear elongated and only the first 480 dots of each line, from the left, will be printed. You will lose the rightmost 180 dots.

If you are using a 72 dot per inch printer, documents will appear only slightly elongated and only the first 572 dots of each line, from the left, will be printed. You will lose the rightmost 88 dots.

For more information on printers and printing, please read the section on Printers below.

In sum, geoPaint is a very powerful graphics editor. A few desirable features are missing, but a few special features have been added. It lacks only absolute coordinates, the ability to set and keep a point, and the ability to draw an image larger than a window to be useful as a true CAD program.

GeoPaint generally works very well, and the few bugs that do occasionally appear are not fatal. GeoPaint will almost never crash, and almost any mistake you make can be recovered if caught immediately.

Desk Accessories

In addition to application programs, GEOS provides Desk Accessories that can be used from deskTop or within applications. These accessories allow you to manipulate Photo and Text Scraps, write notes to yourself, set the time and date for disk file stamping and the alarm clock, change the screen colors, and change the responsiveness of GEOS to your chosen input device.

The first several times you use GEOS, most of the Desk Accessories are not very useful or interesting. You might take a few moments to play with colors, to tell GEOS which printer driver you want to use, etc.

Later, however, you will find the Desk Accessories to be quite useful. In particular, the Photo Manager and the Text Manager will come in handy for moving text and graphics within, around, to, and from geoWrite and geoPaint. GEOS allows only one Text Scrap file and one Photo Scrap file on each disk; the Photo and Text Managers allow you to store many Scraps in Catalogs on disk.

Likewise, the Note Pad will sometimes prove useful for taking quick short notes while you are in the middle of an application, or when you need to leave a message to yourself for your next session. The Note Pad, however, is handicapped by having no print function.

Unless you are really on a strict time schedule, the Alarm Clock is not worth the time it takes to set it. If you diligently set the time, however, GEOS's date and time file stamping can be invaluable. GEOS saves the date and time when a file is updated, and this can help tell several versions of the same document apart. But you must make a habit of setting the time each session for this to be useful.

Desk accessories must be on the current disk in order to be used, but they may be in the border. They may be called from within an application such as geoWrite by using the GEOS/* commands. The accessories will appear under the GEOS/* command in the order they appear on the deskTop notepad. Desk accessories in the border appear last.

As with application programs, desk accessories must be on the current disk. They cannot be placed on the border, the disk closed, and then called when another disk is active.

On extremely rare occasions, using a desk accessory may show a bug within an application. This will only happen when you exceed the capacity of the computer. Desk accessories work by loading over parts of an application, then reloading that part when you close the accessory. If garbage appears anywhere on the screen when you are using, or after you have used, an accessory within an application, exit the application immediately in the normal way. If there is garbage on the deskTop screen, try Special/reset, or completely reboot GEOS. Chances are that you will lose no data - the problem is usually in just a couple of pointers that GEOS uses.

Desk accessories require various amounts of free space on the currently active disk for the reason mentioned above. GEOS packs quite a bit of information into the memory that the C64 allows, and sometimes needs to swap information back and forth to the disk. This should be considered as you work with applications programs and have less than about 25 Kbytes free on disk.

PREFERENCE MANAGER

The Preference Manager allows you to change, set, and save the way GEOS will appear, the way GEOS will respond to a joystick or mouse, and the time and date.

You may select the Preference Manager from anywhere within GEOS, except another desk accessory, by using the GEOS/preference manager command, or by double clicking on the Preference Manager icon. The Preference Manager must be on the currently active disk to be used.

Once the Preference Manager has been selected, GEOS will open a window onto the screen which shows the current settings and in which you can change and save the settings.

In the upper right of the Preference Manager window are five commands. EXIT will return you to deskTop or the application from which you called the Preference Manager. LOAD will get from disk a file named Preferences which contains settings for GEOS other than the default. SAVE will place on disk the settings that are currently being used. Change will activate the changes you have made within the Preference Manager window. For example, you can change the colors all you want, and the Preference Manager window will show those changes in the appropriate box. But you will not see the effect of these changes outside the Preference Manager window until you select Change.

With the exception of Time and Date, all the settings in the Preference Manager will have no effect until you select Change or Save. Selecting Change will change GEOS to the current settings, but will not save the preferences to disk. Selecting Save will save the currently used preferences to disk in a file name Preferences. Changes to the settings in the Preference Manger window will have no effect and cannot be saved unless Change is first selected.

GEOS uses four different color regions - border, background, foreground, and mouse. Each of these regions can be any one of sixteen different colors. That gives 65,536 possible combinations. The DEFAULT command makes it a bit easier to get back to familiar ground if you get lost.

To change the colors, just place the pointer in the color bars at the top center of the Preferences Manager window, and click. The full effect of the different combinations will not be apparent until you select CHANGE, but the border of the color bars will give you a good idea of how different combinations look.

At the upper left of the Preference Manager window are three bars. The first allows you to set the Acceleration, which is how quickly the pointer will change from its minimum to its maximum velocity. The next allows you to set the Maximum velocity, and the last is for Minimum velocity. Basically, the minimum velocity is how fast the pointer will move if you just barely touch the joystick in one direction or another. The maximum velocity is how fast the pointer will move if you hold it joystick in one direction. The default settings are reasonable, but feel free to play with them, especially if you are using a Commodore 1350 type mouse.

To the lower left of the Preference Manager window are two more windows that allow you to redefine the shape of the arrow pointer. When you move the active, real, pointer into the large of these windows, it will change to a tiny two-pixel pointer. You may place this pointer anywhere within the window and toggle the pixels of the pointer image on or off. The effect is shown in the window to the right of the Edit window. These changes do not actually become effective until you select the CHANGE command.

To the lower right of the Preference Manager window are two boxes in which you can set the time and date. These clocks are built into the Commodore 64, but are reset every time you reset the machine. Whenever you save a GEOS file to a GEOS disk, GEOS writes the time and date to the directory entry of the file. This is known as date and time stamping. You can later check the date and time a file was saved by selecting File/info or View/icon.

To set the time, place the pointer anywhere in Time Set box. You may then click, press the space bar, or hit "a" or "p." A space bar will advance you to the next unit to be set - from tens of hours to hours to tens of minutes, etc. When you are on a unit, or the AM/PM, you may only move off the unit by entering a number, the space bar, or <RETURN>. To set the time and start the clock, hit <RETURN>.

For example, the clock currently reads "03:45:27 PM" and I want to set it to match my watch, which is fast approaching 11:43 AM. I move the pointer to Time Set box and enter "A." This will change it from PM to AM and advance me to the tens of hours place. Here I enter a "1," the cursor advances to single hours, I enter "1" again and the cursor moves on to minutes, etc. When I have it set to "11:43:00 AM," I wait for my watch to catch up and then hit <RETURN>. The two timepieces are now synchronized. Moving the pointer out of the Time Set box will do the same as hitting <RETURN>. If the values in a particular location are correct, you may use the space bar to move over them without re-entering the values.

Setting the date is a very similar process, except the cursor starts at the left rather than the right. The Date Set works in MM/DD/YY format.

Unlike most desk accessories, the Exit Button in the Preference Manager does not look like a button, but rather a command. If you select EXIT without first selecting CHANGE, nothing will be changed, except the time and date.

If you make changes, select CHANGE to make the changes effective, and then select SAVE, the Preference Manager will save the currently used settings to a file named Preferences on disk.

The settings in the Preferences file will be loaded and used every time you change disks. For example, if you are using two disk drives, and the disk in device 8 has the Preferences set for a white background and a yellow arrow, while the disk in device 9 has the Preferences set for a blue background and a green arrow, the screen and arrow will change each time you select the other disk drive. Time and date settings are not saved with the Preferences file.

GEOS defaults to 3:00:00 PM, April 7, 1986. The default colors are a black border, light grey background, dark grey foreground, and a dark blue mouse. It is not recommended that you save a Preferences file on the original GEOS boot disk, or attempt to change them on the disk itself.

ALARM CLOCK

The Alarm Clock desk accessory provides a second way of setting the C64's internal clock. It may be called by selecting double clicking on the Alarm Clock icon within deskTop, or by selecting GEOS/alarm clock from deskTop or within an application. In either case, the Alarm Clock must be on the currently active disk.

The Alarm Clock appears in the middle of the screen with the time given in the top half and function buttons in the bottom half. The pointer cannot be moved outside of the Alarm Clock window while it is active.

As you enter the Alarm Clock the first time, the running time will be shown, the cursor will be over the first letter of the AM/PM indicator, and the left button on the bottom half will show a clock face. This indicates that you are looking at the current time as given by the computer's clock. You may change the time by entering the right character in the right place. There are many ways to move around this small window.

For example, the cursor starts at the right of the time. When it is in this position, you cannot enter anything but an "a" or "p," to indicate the half of the day, the space bar to move to the next field. A <RETURN>, "m," or selecting the SET button to set the time. Likewise, the tens of hours position will only accept a one or a two, and if it has a one, the single hours will not accept a three or greater.

You may also use the pointer to place the cursor at a particular position within the clock, you may space bar through the positions of the clock, and you can set the clock by leaving it.

To set the Alarm Clock function, select the clock face on the left to toggle it to a bell. You may then set the alarm in the same manner as the clock, then hit "m," <RETURN>, or select set. This will cause GEOS to sound the alarm at the time you have set. Be careful, however. If you have the alarm set, and then wish to change the time at which the alarm will sound, you must set the alarm twice. Each time you select the clock face, it toggles the alarm switch to the opposite of what it was.

To leave the Alarm Clock and return to deskTop or an application, select the close button on the bottom right, or enter "q."

Setting the time in the Alarm Clock is the same as setting it in the Preference Manager.

NOTE PAD

The Note Pad desk accessory allows you to write notes to yourself and keep them on disk. Each page can have as many as two hundred and fifty-three characters, including carriage returns.

All formatting, except for hard carriage returns are done automatically by the Note Pad. Words will wrap at the end of a line, entering characters will automatically insert before the next character and push the following text, deleting will delete the character to the left of the cursor and pull all the following characters to fill the space.

In no case may more than two hundred and fifty-three characters be entered on any one page - but if fewer characters fill up the page, you may keep on typing beyond the bottom border of the page. The characters will still be there, but invisible and useless unless you delete some of the earlier characters.

GEOS provides for one hundred and twenty-seven pages of notes in the Note Pad which will be stored in a file called NOTES on disk. Only one such file may be on any one disk, and the Note Pad file must be on the disk for it to be used. The NOTES file may not be renamed.

GEOS provides no means for transporting notes from the NOTES file to geoWrite or for printing them independently, for sorting notes, or deleting individual note pages. The whole Notes file, however, may be thrown away.

TEXT MANAGER

The Text Manager desk accessory provides for the collection and storage on disk of Text Scraps created with either geoWrite or geoPaint. GEOS only allows one Text Scrap to be on the disk at any one time, but these Text Scraps can be placed on a page of a Text Album, then later taken out of the Text Album and placed back in the Text Scrap for later use. You may have as many Text Albums on disk as space will allow, a Text Album may have up to one hundred and twenty-seven pages of text taken from Text Scraps, and a scrap may have as many characters as you can fit on full width screen.

You may call the Text Manager from the deskTop by double clicking on its icon, or from within an application or deskTop by selecting the GEOS/text manager command.

The first time you use the Text Manager, you must create a Text Album when prompted. Once created, the Text Manager will present you with a window very similar to the deskTop notepad.

The bottom of the window shows the page number of the album you are looking at, and gives scroll arrows by which you may move the window up and down against the virtual scrap. The main part of the window shows the contents of the page of the album, as drawn from the scrap.

More text than will show on the screen may be in a Text Scrap and on a page in a photo album. When shown in the Text Manager window, the text will wrap to appear attractively in the window, and carriage returns, but not tabs, will be in effect. The scroll arrows at the bottom left can be used to move the window up and down against the text.

The text will always appear in the unenhanced BSW 9 point font. If the original was enhanced or in another font, it will be preserved in the scrap and in the Text Album, but will not show.

At the bottom left of the main part of the window are the page flip dog-ears. Selecting the one to the upper right will cause the Text Manager to flip to the next page of the album, provided it exists. Selecting the one in the lower left will cause it to flip to the previous page, or to the last page, if you are on the first.

There seems to be a definite limit on the size of Text Scrap that the Text Manager will handle. As an extreme test, it will not handle a full window of University 6 point "i"s. It does, however, seem to handle well over a thousand characters quite easily. If the Text Manager locks up, as it may with extremely large scraps, chances are that the scrap is lost and the Text Album corrupt. If you have several Text Manager pages of text in the album, the only way to be certain of recovery is to copy each of the pages into the Text Scrap, then into a geoWrite document. The Text Manager itself seems to lose track when it tries to load an overly large file.

PHOTO MANAGER

The Photo Manager desk accessory provides for the collection and storage on disk of Photo Scraps created with geoPaint. GEOS only allows one Photo Scrap to be on the disk at any one time, but these Photo Scraps can be placed on a page of a Photo Album, then later taken out of the Photo Album and placed back in the Photo Scrap for later use in either geoPaint or geoWrite. You may have as many Photo Albums on disk as space will allow, a Photo Album may have up to one hundred and twenty-seven pages of text taken from Photo Scraps, and a scrap may be as large as one geoPaint drawing screen window.

You may call the Photo Manager from the deskTop by double clicking on its icon, or from within an application or deskTop by selecting the GEOS/photo manager command.

The first time you use the Photo Manager, you must create a photo album when prompted. Once created, the Photo Manager will present you with a window very similar to the deskTop notepad.

The bottom of the window shows the page number of the album you are looking at, and gives scroll box by which you may move the window up and down against the virtual scrap. The main part of the window shows the contents of the page of the album, as drawn from the scrap.

A larger graphic than can be shown on a Photo Manager page can be held within a scrap and within a photo album. When shown in the Photo Manager window, only a portion of the graphic will appear. The scroll box at the bottom left of the Text Manager window can be dragged around the scrap to show the whole thing.

The graphic will always appear as dark grey on light grey. The colors of the original will be preserved in the Photo Scrap and in the Photo Album, but will not show.

At the bottom left of the main part of the window are the page flip dog-ears. Selecting the one to the upper right will cause the Photo Manager to flip to the next page of the album, provided it exists. Selecting the one in the lower left will cause it to flip to the previous page, or to the last page, if you are on the first.

I have found no limit to the size, in bytes, of a Photo Scrap that the Photo Manager can handle, and I suspect that no such limit exists. Be watchful, however, for any garbage on the screen. If and when any appears, back out by normal methods - closing windows, files, exiting applications, etc., and then check as best you can from deskTop.

PRINTING

GEOS is designed for printing on 80 dot per inch printers such as the Epson FX 80 or the Star SG10. If your printer does not print at 80 dots per inch, or an even multiple of 80 dots per inch, then you will not be able to take full advantage of the resolution offered by GEOS application programs.

In geoWrite, this is not too much of a problem. GeoWrite automatically compensates for 72 dot per inch (dpi) and 60 dpi printers by eliminating the automatic left and right margins. You will need to reset the margins to produce an attractive page.

GeoPaint, however, is not so forgiving. The virtual page of a geoPaint document is 640 dots wide by 800 dots down. Assuming that you are using an 80 dpi printer, this allows for an 8 inch by 10 inch graphics page.

Printers with a lower resolution of 72 dpi or 60 dpi cannot compact the dots any closer, and so geoPaint must cut off part of the graphic page instead of printing it. With a 72 dpi printer, the rightmost 64 dots will not be printed. With a 60 dpi printer, the rightmost 160 dots will not be printed. In either case, the printout of your picture will appear elongated, so consider your printer when you are drawing.

If your printer prints 72 dpi or 60 dpi, you are losing nothing by this. GEOS still fully supports your printer, but it supports higher resolution printers as well. Imagine how frustrated you would be if you had an 80 dpi printer, and GEOS would only print three-quarters of the way across the page!

PRINTER DRIVERS

The whole purpose of GEOS is to put things on paper. This is probably the least standardized area of the whole computer industry. There are not more than a two dozen popular operating systems for micro-computers, not that many popular programming languages, and less than half a dozen telecommunication protocols.

GEOS defaults to using the first printer driver that it finds on the disk from which GEOS was loaded. This is no problem if you have a Commodore MPS 1000; if you do not, you must use the GEOS/select printer command to tell it which driver to use. Unless you change to positions of the printer drivers on the boot disk, you must do this every time you load GEOS. It is not recommended that you save files to the original GEOS disk or rearrange the files on it. Rather, make a duplicate boot disk as described in the next section.

There are four major ways of connecting a printer to your computer, and probably one hundred different ways of telling various printers how to print graphics. Further, low-end printers may print a graphic block with as few as 56 positions, and high end printers may use more than four hundred! That is an almost incredible range of variety.

This variety is not too much of a problem if you want to print just simple text. It becomes a bit more difficult if you want such amenities as bold face, double strike, underline, italic, superscript, subscript, or foreign characters. But GEOS, when working with printers, works with graphics. When working with graphics, a program must tell the printer, in the printer's language, where every dot must go on the paper.

Fortunately, most printers nearly follow one of about a half dozen "standards." But this is not standard enough. Different models of the same product line, and even different revisions of the same model, may differ enough - although slightly, still enough - to cause graphics programs to not work properly.

This is not to say that you should go out and buy the most popular printer because it will be supported. Rather, you should find a printer that meets your needs and which is supported by the software you want to use. As long as it conforms fairly closely to any one of the half dozen near standards, you should have no problems for several years.

A printer driver is nothing more than a sort of translation table which a program uses to make the printer print graphics. As different printers have different codes, they need different printer drivers; similar printers can use the same driver almost all the time.

Printers can be connected to a Commodore 64 in four basic ways. Commodore printers, and compatible printers such as the Star-Gemini SL10C or the Seikosha SP 1000VC, connect directly to the Commodore serial port just like a disk drive. These printers should work directly and with few problems when the correct driver is used.

The other three ways of connecting a printer get more complicated in that they require an interface to be used between your computer and the printer. Two of these are not supported by GEOS, and are not likely to be supported in the future - RS232 and IEEE 488. Except for specialized needs, these printers are becoming increasingly rare.

Finally, there are the Centronics type printers such as the Star SG series, the Epson FX and MX series, the Legend series, and others. These printers share about the same popularity among Commodore users as do the direct connect Commodore serial printers.

Centronics type printers require that you use an interface to change the way Commodore sends signals to a way that the printer understands. The Commodore serial standard sends out ones and zeros one at a time. Centronics type printers need to receive the ones and zeros eight at a time. The interface makes this change for you.

Most interfaces do much more than this. Centronics printers all use the same set of codes (known as ASCII) to define characters. In ASCII, an uppercase letter "A" is represented as the number 65 and the lowercase letter "a" is represented by the number 97. The Commodore 64 speaks PETSCII, a strange dialect of ASCII in which an uppercase letter "A" is represented as the number 193 and the lowercase letter "a" is represented by the number 65. Interfaces can translate the letters you send from PETSCII to ASCII. Many interfaces have other special features such as buffers.

When using a Centronics type printer with GEOS, you will usually have two choices. The first choice is to tell your interface that you want it to translate only the serial to the parallel, but not from PETSCII to ASCII. This is because GEOS does all the translation for you. At some times, and with some printers, this may not work properly. If this is the case, you may be able to tell the interface what kind of printer you have, and that it, the interface, is to make that printer act like a Commodore 1525. Then you can use the MPS-1000 printer driver or the MPS-80x printer driver to print from GEOS.

INPUT DRIVERS

An input driver tells GEOS how to respond to a control device such as a joystick or mouse. GEOS V1.2 comes only with a driver for a standard digital joystick. Most mice available for the Commodore machines are digital mice and send the same kind of information to the computer as a joystick does. This makes it possible for you to use the Commodore 1350 mouse, the Contriver C300 (Mighty) mouse, and others.

Since GEOS was marketed, Berkeley Softworks has released at least two drivers for other input devices, namely Inkwell's Flexidraw Lightpen (tm) and Koala Technologies KoalaPad (tm). Neither of these work as well as the original joystick driver, but may be preferable in some cases.

GEOS defaults to using the first input driver that it finds on the disk from which it was loaded. If you wish to use another input driver, you must either select the GEOS/select input command each time you load, or rearrange the order of the input drivers on the disk. It is not recommended that you save files to the original GEOS disk, or rearrange the files on it. Rather, make a duplicate boot disk as described in the next session.

Neither the Flexidraw driver and the Koalapad driver allow you to move as freely as with a joystick or mouse. With both of them, you must first find the pointer before you can move it within the Command menu. This is not too hard with Flexidraw, but can be difficult with a sketchpad as you must match the point on the pad to the correct point on the screen.

Other sketchpads, such as the Suncom Animation Station (tm) can be used with the KoalaPad input driver. Other light pens, however, may need some modification to work properly with the Flexidraw driver. The trigger switch at the tip of the Flexidraw LightPen grounds pin 5 to pin 8 of joyport 0. Such modifications are not recommended, unless you REALLY know what you are doing. A wrong connection can ruin one or more chips in your computer.

Berkeley Softworks has also developed a driver for an analog mouse. Joysticks and digital mice just send an on or off signal to certain pins of the joyport to tell the computer what to do and they have only eight possibilities corresponding to: North, North-east, East, South-east, South, etc. An analog device is not so limited. It can communicate 255 different signals to the computer. This makes it possible to have the computer move the pointer one inch on the screen when you move the mouse one inch on the table. In almost any direction.

Unfortunately, there is not, at this writing, an analog mouse available for the Commodore 64. It is said that one is in development and soon to be released, but no date has been announced.

At least one third party has developed an input driver for GEOS, for the KoalaPad. It is said to ,be more responsive and more accurate. In early October 1986, it was being privately marketed via messages on Quantum Link. Other input drivers are certain to come as programmers dig more deeply into the heart of GEOS.

Berkeley Softworks input drivers are available on Quantum Link, bulletin boards, and from many user's groups.

GENERAL COMMENTS

GEOS is very professionally done, and as nearly bug free as almost any program I have ever worked with. Considering the computer on which it works, it is relatively quick, user-friendly, and versatile. Third party support has been rather weak, as Berkeley Softworks seems reluctant to let out too much information about GEOS while it is still a relatively new and popular product. And, it seems that Berkeley still has many improvements being developed.

But, GEOS V1.2 is, at this time, all we have. It does have bugs, and not all options are always clear. Many of the bugs that occur do not result in a complete system crash requiring that you reset your machine. The few bugs that do crash the system usually involve disk access, and should occur only when you have inadvertently switched disks or opened a drive door.

The first sign of a recoverable bug is usually garbage on the screen. Garbage is any deviation from the normal appearance, and may be a corrupt icon - especially around the name underneath the graphic - or a previous screen showing through or around the current window. If you become aware of a bug, stop and look. DON'T PANIC! There is usually a way out that will not hurt your data.

For example, after several hours of heavy use, garbage may appear around the Status Box in geoPaint. This has happened to me twice, in two different sessions. Both times, I selected File/quit, returned to deskTop, and immediately came back to the same document in geoPaint. The pictures I was working on were uncorrupted.

A similar bug happened several times in geoWrite as I was testing the limit, in characters, of text that a page could handle. As I tried to enter the 5021st character, geoWrite would access the drive and go into an endless loop. There was no way to recover from this without resetting the computer, so I waited several minutes - overnight in one case - just to be sure that it would have a chance to save any data to the disk, and on the chance that it just might be real slow. But, I did indeed have to reset the computer. I then reloaded the same document into geoWrite. All the text came back in, and everything looked good until I tried to do anything. Once again it went into an endless loop. One more time, I loaded the document into geoWrite. This time, the first thing I did was position my cursor at the end of the page (using the scroll box) and delete the last half dozen characters. That solved the problem.

Later, I recreated the same document (which was, by the way, nothing but the lowercase letter "i" in unenhanced University 6 point), deleted a few of the last characters, went back and copied everything into the Text Scrap. Then I tried to copy the Text Scrap onto the first page of a Text Album. I lost the first page of the album. It was simply gone.

As above, most errors that do occur within GEOS are related to the disk. Most notable, and possibly most dangerous, involves unlocking and erasing files on the border. I have corrupted several disks trying to do this. Then again, it sometimes worked properly. But don't trust it.

Finally, try to keep things as simple as you can. GeoWrite will usually go much quicker and cause much less frustration if you worry first about entering text and worry later about making it pretty. GeoPaint can be incredibly frustrating as you try to get the right two colors in each block, only to find that all the colors only make matters worse when you go to print. In some cases, it might be a good idea to work first with the color off (Options/color off), then use the Color command from the Tool Box to go back and paint your picture.

Enjoy!

Programs
&
Utilities

IMPORTING TEXT

While geoWrite is an excellent program for printing attractive pages, it is somewhat lacking in other word processing features, most notably speed.

Many users find it desirable to use another word processor in order to take advantage of speed, spelling checkers, search and replace functions, etc.

The following utility, with the appropriate module, will allow you to import text from a few of the more popular word processors into geoWrite. Once within geoWrite, the text can be reformatted, the fonts changed, and enhancements and graphics added before you finally print your text. Though not ideal, this utility allows you to take some advantage of both worlds.

The principle of the text conversion utility is simple. GEOS handles text in a way very similar to other word processors; the main difference is in the file structure. The GEOS file structure is explained in the last section of the book.

The programmers of this utility went to some length to allow for any file from the supported word processors to be imported without modification. We could not, however, explore every possibility. If you have any problems with this utility, please remove formatting commands, text enhancement commands, redefined characters, and the more unusual characters (British pound sign, up arrow, pi, etc.) from the original text file. Most important, and most likely to cause trouble, are header and footer commands. These commands use unusual syntax and may cause the conversion program to steal characters from your file.

The programs are easy to use and nearly self-explanatory. You need only copy the original file to a GEOS disk, run the utility, and follow the prompts. The utility will read the original file, translate it into GEOS compatible text, and build a GEOS file. The original file will remain unaltered.

The listing for the BASIC program to convert an Easy Script file to a geoWrite file is given below. The listing is printed as it will appear in lower case/upper case mode on the Commodore 64 or the forty column screen of the Commodore 128. Immediately following the listing of the Easy Script conversion are replacement lines to be used for SpeedScript, Pocket (Paperback) Writer, PaperClip, and Word Pro. Finally, source code for the machine language routines is given at the end.

KEY

Some of the characters in this, and following, listings, appear odd. Most of these are screen control characters, but a few are control codes which GEOS uses for internal information. All odd characters will appear within quotes, and, with two exceptions, are similar to how they will appear in lowercase/uppercase mode on the screen.

The greek letter mu (a ”u” with a tail on its left side) represents the <logo><asterisk>. To enter this character, press and hold down the Commodore logo key at the lower left of the keyboard, then press and release the asterisk, near the <RETURN> key. This will appear on your screen as a triangle in the upper right half of the character block.

The underlined lower_case ”a” (a) can be entered by pressing and holding down the <CONTROL> key and pressing the <a> key. This will appear on your screen as a reversed ”a.”

The remainder of the characters are screen codes. All are underlined, and may be entered within quotes as follows:

S	<SHIFT><CLR/HOME>
s	<CLR/HOME>
q	<cursor down>
Q	<cursor up>
r	<CONTROL><9> (reverse on)
R	<CONTROL><0> (reverse off)
]	<cursor right>
1/4	<cursor left>

Most spaces within the program are cosmetic, with the exception of lines containing ”print#15,” as in the following example:

```
320 print#15,"u1:2 0 18 0"
```

in which case all spaces must be maintained or replaced by commas.

Note also that the BASIC "print#" command may NOT be abbreviated with a "?#." If you wish to abbreviate this command, use "pR" (<p> then a <SHIFT><R>).

All REM statements may be removed.

Type in the program and save it to disk, then copy it to a GEOS disk and try it. Be sure to save a copy of the program you type in.

The first several times you use any of these programs, use them only on a work disk which has no valuable data. Use a copy. These programs write directly to the disk, and although every effort has been made to make them bug free, we cannot guarantee your typing. Make sure the programs work before you risk anything but a bit of time!

The conversion program for Easy Script can be modified to work with other word processors by simply replacing lines 11000 through 11900 with the appropriate modules that follow.

```
100 rem**********************************
****************************************
110 rem***** easy script to geos conversi
on program - s. gast & r. rollins ******
120 rem**********************************
****************************************
130 ifff=1thensys24576:rem reboot geos
140 forx=49152to 49490:rem install ml
150 read d
160 poke x,d
170 nextx
180 gs$="geos "+chr$(102)+chr$(111)+chr$(
114)+chr$(109)+chr$(97)+chr$(116)
190 pd$="                ":ps$="
       ":nu$=chr$(0)
200 n$=chr$(0):h$=n$+n$+"µa("+n$+"µaµaµaµ
aµaµaµawi"+n$+n$:rem geos page headder
210 printchr$(14)
220 poke53280,0:poke53281,15:poke646,11:r
em change screen & character colors
230 print"Sqqqqq  Insert disk with Word P
rocessor file "
240 print"q    on it into disk drive numb
er 8"
250 print"q       Press any key to contin
ue"
260 geta$:if a$="" then 260
270 print"qqq            r Please Wait R"

280 open15,8,15,"i0":rem initialize the d
isk drive
290 gosub 1800:rem check the error channe
l
300 open2,8,2,"#":rem prepare direct acce
ss channel buffer
310 gosub 1800
320 print#15,"u1:2 0 18 0":rem bring trac
k 18 sector 0 into the drive buffer
```

```
330 gosub 1800
340 print#15,"b-p:2 173":rem position poi
nter to start of geos id name
350 gosub 1800
360 b$="":x=0
370 gosub 1920:rem get a byte from channe
l number 2
380 b$=b$+a$:rem concat the string to for
m geos id name
390 x=x+1
400 if x<11 then 370:rem get 11 character
s
410 if b$<>gs$ then 1880:rem make sure it
's a geos disk
420 print"Sqqqqqqq"
430 poke198,0:fc$="":fl$=""
440 input"Please enter file name (16 lett
ers only)";fl$
450 if len(fl$)<1or len(fl$)>16thenprint"
Sqqqqqqq              try againqq":goto4
40
460 fc$=right$(pd$,(16-len(fl$)))
470 fc$=fl$+fc$
480 print"Sqqqqqqq       Is this the cor
rect file?"
490 print"qq          r "fc$" R"
500 print"qq               Enterq":print
"   'y' for yes 'n' for no 'q' for quit"
510 poke198,0
520 geta$:if a$=""then520
530 if a$="n"then 420
540 if a$="q" then1850
550 if a$<>"y"then520
560 close2
570 print"sqqqqqqqqqqqqqqqqqqqq
    r WORKING R      "
580 open2,8,2,"0:"+fl$
590 input#15,en,em$,et$,es$:rem check the
 error channel
600 if en=62then 2260
610 close2
```

```
620 print#15,"m-r"chr$(144)chr$(2)
630 get#15,ds$:ifds$=""thends$=chr$(0)
640 ds=asc(ds$)
650 print#15,"m-r"chr$(148)chr$(2)
660 get#15,dp$:ifdp$=""thendp$=chr$(0)
670 dp=asc(dp$)
680 print#15,"m-r"chr$(250)chr$(2)
690 get#15,a$:ifa$=""thena$=chr$(0)
700 bf=asc(a$)
710 print#15,"m-r"chr$(252)chr$(2)
720 get#15,a$:ifa$=""thena$=chr$(0)
730 bf=bf+(asc(a$)*256)
740 open2,8,2,"#"
750 gosub 1800
760 print#15,"u1:2 0 18"ds
770 gosub1800
780 print#15,"b-p:2"(dp+28)
790 gosub1800
800 get#2,a$:if a$=""thena$=chr$(0)
810 fs=asc(a$)
820 get#2,a$:ifa$=""then a$=chr$(0)
830 fs=fs+(asc(a$)*256)+5
840 ifbf-fs<0 then 2170
850 close2
860 print"sqqqqqqqqqqqqqqqqqqqq
     WORKING   "
870 open3,8,3,"0:test.s.e.q,s,w"
880 gosub1800
890 print#3,"steve and rich";
900 close3
910 sys49161
920 open3,8,3,"0:test.s.e.q,s,r"
930 gosub1800
940 print#15,"m-r"chr$(144)chr$(2)
950 get#15,ds$:ifds$=""thends$=chr$(0)
960 ds=asc(ds$)
970 print#15,"m-r"chr$(148)chr$(2)
980 get#15,dp$:ifdp$=""thendp$=chr$(0)
990 dp=asc(dp$)
1000 close3
1010 gosub 2310
```

```
1020 print#3,nu$;
1030 gosub 2230
1040 dt$=a$:dt=asc(dt$)
1050 gosub 2230
1060 sd$=a$:sd=asc(sd$)
1070 print#15,"u2:3 0 18"ds
1080 close3
1090 print"sqqqqqqqqqqqqqqqqqqqq
     r WORKING R      "
1100 open3,8,3,"0:test.s.e.q,s,w"
1110 gosub1800
1120 forx=1to254
1130 read d:print#3,chr$(d);
1140 next x
1150 close3
1160 gosub 2310
1170 print#3,nu$;
1180 gosub2230
1190 it$=a$:it=asc(it$)
1200 gosub2230
1210 is$=a$:is=asc(is$)
1220 print#15,"u2:3 0 18"ds
1230 close3
1240 open2,8,2,"0:"+fl$
1250 gosub1800
1260 open3,8,3,"0:test.s.e.q,s,w"
1270 gosub1800
1280 gosub2390
1290 print#3,h$;
1300 sys49152
1310 close3
1320 ck=peek(2):ifck<>0then 1950
1330 gosub2310
1340 print#3,nu$;
1350 gosub2230
1360 track=asc(a$)
1370 gosub2230
1380 sectr=asc(a$)
1390 print#15,"u2:3 0 18"ds
1400 close3
1410 poke780,track:poke781,sectr
```

```
1420 sys49155
1430 ck=peek(2):ifck<>0then 1950
1440 fl=peek(251):if fl=0 then1260
1450 gosub 2310
1460 gosub 1640
1470 print#15,"u2:3 0 18"ds
1480 gosub1800
1490 print#15,"u1:3  0"dt;sd
1500 gosub1800
1510 sys49158
1520 print#15,"u2:3 0"dt;sd
1530 gosub 1800
1540 print"Sqqqqqqq           r PROGRAM C
OMPLETE R"
1550 print"qq         r FILE IS"t"r¼ PAGE
S LONG R"
1560 print"qq              Happy to help"

1570 print"qq        Press 'g' to return 
to Geos"
1580 print"qq          any other key to q
uit"
1590 getrr$:ifrr$="" then 1590
1600 if rr$<>"g" then 1850
1610 goto 2430
1620 end
1630 :
1640 print#15,"m-r"chr$(250)chr$(2)
1650 get#15,a$:ifa$=""thena$=chr$(0)
1660 bs=asc(a$)
1670 print#15,"m-r"chr$(252)chr$(2)
1680 get#15,a$:ifa$=""thena$=chr$(0)
1690 bs=bs+(asc(a$)*256)
1700 bc=bf-bs:bh=int(bc/256):bl=bc-(bh*25
6)
1710 ac$=""
1720 forx=1tolen(fl$):b$=mid$(fl$,x,1):if
 b$="*"thenx=len(fl$):goto1740
1730 b=asc(b$):gosub2120:ac$=ac$+b$:nextx

1740 ac$=ac$+right$(ps$,(16-len(ac$)))
```

```
1750 di$=chr$(131)+dt$+sd$+ac$+it$+is$+ch
r$(1)+chr$(7)+chr$(86)+chr$(10)
1760 di$=di$+chr$(4)+chr$(1)+chr$(0)+chr$
(bl)+chr$(bh)
1770 print#3,di$;
1780 return
1790 :
1800 input#15,en,em$,et$,es$
1810 if en<20 then return
1820 print"Sqqqqqqq       r A disk error
 has happened R"
1830 print"    Please rerun program and t
ry again"
1840 print"q          "en;em$" "et$" ";es
$
1850 close2:close3:close15
1860 end
1870 :
1880 print"Sqq         This is not a Geos
 disk"
1890 print"Please rerun program and use a
 Geos disk"
1900 goto 1850
1910 :
1920 get#2,a$:if a$=""then a$=chr$(0)
1930 return
1940 :
1950 print"Sqqqqq         This is a corrup
ted file"
1960 print"q       Program will not work
on it"
1970 print"q    Please rewrite file and t
ry again"
1980 print"q          ]Press 'q' to qui@
$1
1990 print"q       Any other key to go to
 Geos"
2000 geta$:if a$=""then2000
2010 if a$<>"q"then2430
2020 goto 1850
2030 print"Sqqqqq         r "fc$" R"
```

```
2040 print" q is not on this disk PLEASE
 try again"
2050 print"qq        Enter 'y' to try ag
ain"
2060 print"q        Any other key to qui
t"
2070 poke198,0
2080 geta$:ifa$=""then2080
2090 if a$<>"y"then1850
2100 close2:close3:close15:goto190
2110 :
2120 if b>64and b<91 thenb=b+32 :b$=chr$(
b)
2130 if b>192andb<224thenb=b-96 :b$=chr$(
b)
2140 return
2150 printb;asc(b$):return
2160 :
2170 print"Sqqqq There is not enough roo
m on this disk"
2180 print"q   Please transfer file to a
nother"
2190 print"q      Geos disk with more sp
ace":goto1980
2200 print"Sqqqqqqq           r error R"

2210 goto1850
2220 :
2230 get#3,a$:if a$="" then a$=chr$(0)
2240 return
2250 :
2260 print"Sqqqq]]]]]]The file is not on
 this disk"
2270 print"q]]]]Please insert disk with t
he file"
2280 print"q]]]]into disk drive #8 and tr
y again"
2290 goto2050
2300 :
2310 open3,8,3,"#"
2320 gosub1800
```

```
2330 print#15,"u1:3 0 18"ds
2340 gosub1800
2350 print#15,"b-p:3"dp
2360 gosub1800
2370 return
2380 :
2390 t=t+1
2400 print"sqqqqqqqqqqqqqqqqqqqq        r
 MAKING PAGE NUMBER R     "t
2410 return
2420 :
2430 close2:close3:close15
2440 poke53272,(peek(53272)and15)or192:po
ke648,48
2450 print"S"
2460 open15,8,15:open2,8,2,"geos boot,p,r
":input#15,gg:close2:close15
2470 ifgg=0then2520
2480 print"sqqqqXnInsert Geos Disk in Dri
ve 0."
2490 print"Hit any key to continue..."
2500 geta$:ifa$=""then2500
2510 goto2450
2520 ifff=0thenff=1:load"geos boot",8,1
2530 end
9999 rem universal ml module
10000 data76,170,192,76,109,192,76,127,19
2,76,147,192,0,0,2,0,169,0,133,2,32
10100 data204,255,162,3,32,201,255,169,12
,32,210,255,32,204,255,96,162,2,32
10200 data198,255,32,207,255,166,144,208,
6,72,32,204,255,104,96,72,169,255,133
10300 data251,32,204,255,104,96,72,162,3,
32,201,255,104,72,32,210,255,32,204
10400 data255,104,205,169,192,240,10,238,
13,192,173,13,192,201,79,144,8,169,0
10500 data141,13,192,238,12,192,96,169,25
5,133,2,96,172,14,192,153,0,207,200
10600 data138,153,0,207,200,240,188,140,1
4,192,96,162,3,32,201,255,160,0,185,0
10700 data207,32,210,255,200,208,247,32,2
```

```
04,255,96,162,0,169,0,157,0,207,232
10800 data169,255,157,0,207,232,208,242,1
69,255,141,15,192,96
10900 rem********************************
**************************************
10999 rem easy script ml module
11000 data13,173,15,192
11100 data240,3,32,36,193,169,0,141,15,19
2,133,144,133,2,133,251,141,13,192
11200 data141,12,192,165,251,240,1,96,32,
37,192,201,128,240,40,201,65,144,20
11300 data201,91,176,5,105,32,76,231,192,
201,92,240,227,201,95,144,3,56,233
11400 data128,32,65,192,173,12,192,201,60
,176,3,76,195,192,76,16,192,32,37,192
11500 data72,169,251,208,4,104,76,55,192,
104,201,42,240,2,208,17,32,37,192,72
11600 data169,251,208,4,104,76,55,192,104
,201,13,208,3,76,195,192,201,59,240
11700 data249,208,230,96,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
11800 data0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0
11900 rem********************************
**************************************
14999 rem info block data
15000 data 3,21,191
15100 data255,255,255,128,0,1,136,128,1,1
33,117,97,130,85,65,130,119,65,128,0
15200 data1,143,81,193,136,81,1,142,81,19
3,136,81,1,136,93,193,128,0,1,255,255
15300 data255,0,0,1,247,119,17,132,84,49,
151,87,127,148,81,48,247,119,16,0,0,0
15500 data131,7,1,0,0,255,255,0,0,87,114,
105,116
15600 data101,32,73,109,97,103,101,32,86,
49,46,49,0,0,0,0
15610 data83,46,32,71,97,115,116,32,38,32
,82,46,32,82,111,108,108,105,110,115
15700 data103,101,111,87,114,105,116,101,
32,32,32,32,86,49,46,49,0
```

```
15800 data0,0,0,162,16,189,124,40,157,107
,40,202,208,247,32,75,193,169,0,133,46
15900 data32,60,57,169,40,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
16000 data0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
16100 data0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
```

SPEEDSCRIPT MODULE

```
11000 data13,173,15,192
11100 data240,3,32,249,192,169,0,141,15,1
92,133,144,133,2,133,251,141,13,192
11200 data141,12,192,165,251,240,1,96,32,
37,192,48,246,240,59,201,27,144,61
11300 data201,28,144,51,240,234,201,31,14
4,45,240,55,201,64,144,6,240,222,201
11400 data91,176,50,32,65,192,173,12,192,
201,60,176,3,76,195,192,76,16,192,162
11500 data2,32,198,255,160,2,32,207,255,1
36,208,250,32,204,255,96,24,105,64,76
11600 data233,192,24,105,96,76,233,192,16
9,13,76,233,192,76,104,192,0,0,0,0,0
11700 data0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
11800 data0,0,0,0,0,0,0,0,0,0,0,0,0,0
```

POCKET (PAPERBACK) WRITER MODULE

```
11000 data13,173,15,192
11100 data240,3,32,253,192,169,0,141,15,1
92,133,144,133,2,133,251,141,13,192
11200 data141,12,192,165,251,240,1,96,32,
37,192,240,65,201,122,240,105,201,27
11300 data144,63,240,55,201,29,144,232,24
0,49,201,31,144,45,240,55,201,64,144
11400 data6,240,218,201,91,176,214,32,65,
192,173,12,192,201,60,176,3,76,195
11500 data192,76,16,192,162,2,32,198,255,
160,62,32,207,255,136,208,250,32,204
11600 data255,96,24,105,64,76,237,192,24,
105,96,76,237,192,169,13,32,65,192
11700 data173,12,192,201,60,176,212,32,37
,192,201,32,240,249,72,169,32,32,65
11800 data192,104,76,203,192,76,104,192,3
2,37,192,201,31,240,4,201,59,208,245
11900 data76,195,192,245,76,195,192,32,37
,192,76,195,192,2
```

PAPERCLIP MODULE

```
11000 DATA13,173,15,192
11100 DATA240,3,32,1,193,169,0,141,15,19
2,133,144,133,2,133,251,141,13,192,141
11200 DATA12,192,165,251,240,1,96,32,37,
192,240,69,201,122,240,109,201,30,240
11300 DATA119,201,27,144,63,240,55,201,2
9,144,228,240,49,201,31,144,222,240,55
11400 DATA201,64,144,6,240,214,201,90,17
6,210,32,65,192,173,12,192,201,60,176
11500 DATA3,76,195,192,76,16,192,162,2,3
2,198,255,160,62,32,207,255,136,208
11600 DATA250,32,204,255,96,24,105,64,76
,241,192,24,105,96,76,241,192,169,13
11700 DATA32,65,192,173,12,192,201,60,17
6,212,32,37,192,201,32,240,249,72,169
11800 DATA32,32,65,192,104,76,203,192,76
,104,192,32,37,192,201,31,240,4,201,59
11900 DATA208,245,76,195,192,32,37,192,7
6,195,192,0
```

WORDPRO MODULE

```
11000 DATA13,173,15,192,240,3,32,1,193,1
69,0,141,15,192,133,144,133,2,133,251
11100 DATA141,13,192,141,12,192,165,251,
240,1,96,32,37,192,240,69,201,122,240
11200 DATA114,201,255,240,73,201,27,144,
63,240,55,201,29,144,228,240,49,201,31
11300 DATA144,222,240,60,201,64,144,6,24
0,214,201,91,176,210,32,65,192,173,12
11400 DATA192,201,60,176,3,76,195,192,76
,16,192,162,2,32,198,255,160,118,32
11500 DATA207,255,136,208,250,32,204,255
,96,24,105,64,76,241,192,24,105,96,76
11600 DATA241,192,169,32,76,241,192,169,
13,32,65,192,173,12,192,201,60,176,207
11700 DATA32,37,192,201,32,240,249,72,16
9,32,32,65,192,104,76,203,192,76,104
11800 DATA192,32,37,192,201,31,240,4,201
,59,208,245,76,195,192,0,0
```

IMPORTING GRAPHIC FILES

Unlike text files, geoPaint is as adequate as most standard graphics program for creating pictures. There is, however, a vast library of graphics files available in the public domain and the ability to use these pictures in full page geoPaint and geoWrite documents can only enhance the value of both GEOS and the available art.

The following utility will allow you to translate a graphic file from Doodle! (tm), Flexidraw (tm), or Koala Pad (tm) into a nearly standard compacted geoPaint file. Each of the three graphics programs work with a screen slightly larger than the geoPaint working window, so you will not be able to see the complete picture from one window. Moving the window against the background of the virtual document page will allow you to see the complete picture. Unfortunately, a complete graphic from one of the other programs will not fit into a Photo Scrap, and so the complete graphic cannot be used in geoWrite.

The program is very easy to use. The utility completely loads the Doodle!, Flexidraw, or Koala Pad graphic file into the computer from the source disk, translates it, and then saves it to the target disk. The source and target disk do not have to be the disk, allowing you to have a complete disk dedicated to your graphics library and use your normal GEOS work disks as targets.

The utility does require that the target disk have at least 50 blocks free on the disk. This is a rough estimate of the maximum size a compacted graphic could occupy. A file this size would be extremely rare, and in general use, the geoPaint file created by this utility will be substantially smaller than the source file.

To use this utility, simply load it in, run it, and follow the prompts. Graphics file flag prefixes will be handled by the utility. For example, you do not need to type in the leading "DD" which prefixes a Doodle! filename, nor the reversed spade character [CHR$(129)] which prefixes Koala Pad filenames.

Suffixes, however, must be included in the filename. The filename should be entered as it appears in a standard Commodore BASIC directory listing in lowercase/uppercase mode.

Once you given a valid name and the file is found on disk, the utility will display a high resolution screen as the file is loaded in. Flexidraw are normally black and white, and no provision has been made for color Flexidraw pictures, so the display and the geoPaint file will default to standard geoPaint dark grey on light grey.

Doodle! files use color, and so the original colors will be displayed when the Doodle! file is first loaded. You will then have the option of changing the file of the Doodle! picture to the geoPaint standard dark grey on light grey. THIS IS NOT RECOMMENED! Once you change the color to the default, you cannot recover the original colors without starting over. The colors will have no adverse effect on geoPaint, and will display normally. If you would rather work in monochrome, select Options/color off from the geoPaint command menu.

Koala Pad files are multi-color, as opposed to high resolution, graphics. Multi-color graphics allow you to use four colors per eight pixel by eight pixel graphics card instead of the two colors allowed by high resolution. To do this, the Commodore 64 sacrifices horizontal resolution, providing 160 horizontal dots of resolution as opposed to the 320 horizontal dots provided by high resolution mode.

The end effect of this is that many Koala Pad pictures will not translate well into high resolution pictures without dithering, which can get quite complex. With translation of multi-color Koala Pad pictures into high resolution pictures, you will have the option of preserving foreground and background colors, multi-color 1 and multi-color 2, or setting the colors to the geoPaint default of dark grey on light grey. As with Doodle! pictures, once the geoPaint default colors have been selected, the original foreground and background colors cannot be recovered. Experiment with the colors, and be prepared to accept that many Koala Pad pictures will not be worth translating into high resolution graphics. Another option, but beyond the scope of this book would be to completely rewrite the Koala Pad picture to take advantage of the 640 horizontal dots of resolution offered on the geoPaint document page. Such a utility may appear either in the commercial market or in the public domain.

Doodle!, Flexidraw, and Koala Pad all store there files in a nearly standard format. None of them make any attempt at data compression. GeoPaint, on the other hand, uses a very effective method of data compression and a file structure that makes it easier to work with a virtual page that is much larger than the screen. This utility compresses the original graphic data and builds a standard GEOS VLIR file. The color data is handled a bit more loosely, as there is so little of it. For a complete description of geoPaint file structure and data structure, please refer to section three.

The complete utility is listed, ready to type in and run, below. It includes machine language routines as data statements. The source code, in PAL (tm) format, is given after the program listing. For an explanation of special characters, please refer to the KEY given on page 2-3.

As with any program, type-in or otherwise, do not risk anything the first few times you run it. Once you have entered the program, save it to disk, then run it using an unimportant target disk. The utility writes directly to the disk, and if it does not run properly can corrupt several files. Be careful the first several times you use it.

If you encounter any problems, check your typing, your source disk, and your target disk. Remember that the target disk must have at least 50 blocks (about 13 Kbytes) free.

```
10 rem "****************************
20 rem "**** convert to geoPaint *****
30 rem "****     r. rollins      *****
40 rem "****        and          *****
50 rem "****      s. gast        *****
60 rem "****      (c) 1986       *****
70 rem "****    Midnite Press    *****
80 rem "** Midnite Software Gazette **
90 rem "****************************
95 :
1000 onbgoto1300,1390,3560
1010 poke56,92:clr:poke53281,15:poke53280
,0:printchr$(14):poke646,11
1020 open15,8,15
1030 gosub1570
1040 print"Sq]WWhat type of picture do yo
u wish to"
1050 print"]convert to a geopaint file?"
1060 print"qq]]]]]]]]]]]]1 = Doodle
1070 print"q]]]]]]]]]]]]2 = Flexi draw
1080 print"q]]]]]]]]]]]]3 = Koala
1090 print"qqq]Please select the number o
f your choice"
1100 getw$:ifw$<>"1"andw$<>"2"andw$<>"3"t
hen1100
1110 g=val(w$):onggoto1120,1140,1150
1120 print"q]eEnter the filename excludin
g the DD"
1130 print"]that precedes the picture's n
ame:":goto1170
1140 print"q]eEnter the filename:":goto11
70
1150 print"qeEnter the filename excluding
 the reverse";
1160 print"spade that precedes the pictur
e's name:"
1170 input"Wq]]]]]]]";f$
1180 fl$=f$:iff$=""then1030
1190 ifg=1thenf$="dd"+f$:gosub1950:goto12
20
```

```
1200 ifg=2thengoto1220
1210 ifg=3thengosub1790
1220 poke53265,peek(53265)or32:rem hi-res
 on
1230 poke56578,peek(56578)or3:rems cia #2
 bits 0 and 1 to output
1240 poke56576,(peek(56576)and252)or2:rem
 selects video bank 1 ($4000-$7fff)
1250 poke53272,(peek(53272)and15)or120:re
m screen offset within bank ($2000)
1260 poke648,96:rem page of screen (for k
ernal screen editor)
1270 ifg=2then1360
1280 open2,8,2,"0:"+f$:gosub1550:close2
1290 b=1:loadf$,8,1
1300 onggoto1320,1360,1430
1310 poke198,0
1320 getw$:ifw$=""then1320
1330 ifw$="b"thenforx=23552to24551:pokex,
191:nextx
1340 forx=1to2000:next
1350 goto1520
1360 open2,8,2,""+f$+",p,r"
1370 poke185,0:poke780,0:poke781,0:poke78
2,96
1380 b=2:sys62622
1390 gosub1550:close2
1400 forx=23552to24551:pokex,191:next
1410 forx=1to2000:nextx
1420 goto1520
1430 forx=23552to24551:pokex,peek(x+8000)
:nextx:goto1470
1440 forx=23552to24551:pokex,peek(x+9000)
:nextx:goto1470
1450 forx=23552to24551:pokex,191:nextx
1460 poke198,0
1470 getw$:ifw$=""then1470
1480 ifw$="1"then1430
1490 ifw$="2"then1440
1500 ifw$="b"then1450
1510 goto1520
```

```
1520 poke53265,27:poke53272,23:poke648,4:
poke56576,151:rem back to default
1530 ifmm=1thenprint"Sqq"d,e$,f,h:close2:
close15:end
1540 goto2050
1550 input#15,d,e$,f,h:ifd<20thenreturn
1560 mm=1:goto1520
1570 print"WSqq]]]Do you wish to see a di
sk directory?"
1580 print"]]]]]]]]]]]]]]]]]qqqqqY = yes"

1590 print"]]]]]]]]]]]]]]]]]qN = no"
1600 print"qq]]]]]]]]]Please enter your c
hoice"
1610 getw$:ifw$<>"n"andw$<>"y"then1610
1620 print"S"
1630 ifw$="n"thenreturn
1640 open1,8,0,"$0"
1650 get#1,a$,a$
1660 get#1,a$,a$
1670 ifa$=""then1740
1680 get#1,a$,b$
1690 printasc(a$+chr$(0))+asc(b$+chr$(0))
*256;
1700 get#1,a$
1710 ifa$=""thenprint:goto1660
1720 printa$;
1730 goto1700
1740 close1:close8
1750 print"qPress any key to continue"
1760 poke198,0
1770 getw$:ifw$=""then1770
1780 return
1790 print"S"chr$(17)"q]The picture will
be displayed in"
1800 print"]two of the original Koala col
ors":print
1810 print"]Press 2 to toggle to the othe
r two"
1820 print"]colors of the original multic
olor"
```

```
1830 print"]screen":print
1840 print"]press B to place in geos defa
ult"
1850 print"]colors (in essence black & wh
ite)":print
1860 print"]Press 1 to return to the orig
inal two"
1870 print"]colors selected":print
1880 print"]Any other key will begin the
file"
1890 print"]conversion.":print:print
1900 print"     Press any key to continue
"
1910 poke198,0
1920 getw$:ifw$=""then1920
1930 f$=chr$(129)+f$:l=len(f$):l=15-l:for
x=1tol:m$=m$+" ":nextx:f$=f$+m$
1940 return
1950 print"S"chr$(17)"q]The picture will
be displayed in"
1960 print"]the original colors, if any."
:print
1970 print"q]press B to place in geos def
ault"
1980 print"]colors (in essence black & wh
ite)"
1990 print"]press any other key to keep t
he "
2000 print"]original color selection.":pr
int:print
2010 print"     Press any key to continue
"
2020 poke198,0
2030 getw$:ifw$=""then2030
2040 return
2050 print"Sqqqqqqqqq]Please insert a  G
EOS  disk that you"
2060 print"]want the Geopaint file to be
made on"
2070 print"]]qqq     Press any key when r
eady"
```

```
2080 poke198,0
2090 getw$:ifw$=""then2090
2100 close15
2110 print"Sqqqqqqq]Please wait while th
e ml is installed..
2120 for x =49152 to 49719:readd:pokex,d:
nextx
2130 print"Sqqqqqqq]Now creating the GEO
S file...."
2140 pd$="                ":ps$="
        ":nu$=chr$(0)
2150 printchr$(14)
2160 poke53280,0:poke53281,15:poke646,11:
poke2,0:poke808,234
2170 open15,8,15,"i0"
2180 gosub3190
2190 open2,8,2,"#"
2200 gosub 3190
2210 print#15,"u1:2 0 18 0"
2220 gosub 3190
2230 print#15,"b-p:2 173"
2240 gosub 3190
2250 b$="":x=0
2260 gosub 3380
2270 b$=b$+a$
2280 x=x+1
2290 if x<11 then 2260
2300 if asc(b$)<>asc("geos FORMAT")then32
70
2310 print#15,"m-r"chr$(250)chr$(2)
2320 get#15,a$:ifa$=""thena$=chr$(0)
2330 bf=asc(a$)
2340 print#15,"m-r"chr$(252)chr$(2)
2350 get#15,a$:ifa$=""thena$=chr$(0)
2360 bf=bf+(asc(a$)*256)
2370 if bf-50<0 then 3330
2380 open3,8,3,"test.s.e.q,s,w"
2390 gosub 3190
2400 print#3,"steve and richard"
2410 close3
2420 sys 49155 :rem initilize
```

```
2430 open3,8,3,"0:test.s.e.q,s,r"
2440 gosub3190
2450 print#15,"m-r"chr$(144)chr$(2)
2460 get#15,ds$:ifds$=""thends$=chr$(0)
2470 ds=asc(ds$)
2480 print#15,"m-r"chr$(148)chr$(2)
2490 get#15,dp$:ifdp$=""thendp$=chr$(0)
2500 dp=asc(dp$)
2510 close3
2520 gosub 3400
2530 print#2,nu$;
2540 gosub 3380
2550 dt$=a$:dt=asc(dt$)
2560 gosub 3380
2570 sd$=a$:sd=asc(sd$)
2580 print#15,"u2:2 0 18"ds
2590 open3,8,3,"0:test.s.e.q,s,w"
2600 gosub3190
2610 forx=1to254
2620 read d:print#3,chr$(d);
2630 next x
2640 close3
2650 gosub 3400
2660 print#2,nu$;
2670 gosub3380
2680 it$=a$:it=asc(it$)
2690 gosub3380
2700 is$=a$:is=asc(is$)
2710 print#15,"u2:2 0 18"ds
2720  open3,8,3,"0:test.s.e.q,s,w"
2730 gosub3190
2740 sys49152
2750 sg=sg+1:print"s"sg
2760 close3
2770 gosub 3400
2780 print#2,nu$;
2790 gosub 3380
2800 track=asc(a$)
2810 gosub 3380
2820 sectr=asc(a$)
2830 print#15,"u2:2 0 18"ds
```

```
2840 poke780,track:poke781,sectr
2850 sys49158
2860 fl=peek(2):if fl=0 then 2720
2870 gosub 3400
2880 gosub 3040
2890 print#15,"u2:2 0 18"ds
2900 gosub3190
2910 print#15,"u1:2  0"dt;sd
2920 gosub3190
2930 sys49161
2940 print#15,"u2:2 0"dt;sd
2950 gosub 3190
2960 print"Sqqqqqqq           r PROGRAM C
OMPLETE R"
2970 print"qq              Happy to help"

2980 print"qq        Press 'g' to return
to Geos"
2990 print"qq          any other key to q
uit"
3000 getrr$:ifrr$="" then 3000
3010 if rr$<>"g" then 3240
3020 goto 3450
3030 end
3040 print#15,"m-r"chr$(250)chr$(2)
3050 get#15,a$:ifa$=""thena$=chr$(0)
3060 bs=asc(a$)
3070 print#15,"m-r"chr$(252)chr$(2)
3080 get#15,a$:ifa$=""thena$=chr$(0)
3090 bs=bs+(asc(a$)*256)
3100 bc=bf-bs:bh=int(bc/256):bl=bc-(bh*25
6)
3110 ac$=""
3120 forx=1tolen(fl$):b$=mid$(fl$,x,1):if
 b$="*"thenx=len(fl$):goto3140
3130 b=asc(b$):gosub3300:ac$=ac$+b$:nextx

3140 ac$=ac$+right$(ps$,(16-len(ac$)))
3150 di$=chr$(131)+dt$+sd$+ac$+it$+is$+ch
r$(1)+chr$(7)+chr$(86)+chr$(10)
3160 di$=di$+chr$(4)+chr$(1)+chr$(0)+chr$
```

```
(bl)+chr$(bh)
3170 print#2,di$;
3180 return
3190 input#15,en,em$,et,es
3200 if en<20 then return
3210 print"Sqqqqqqq       r A disk error
 has happened R"
3220 print"   Please rerun program and t
ry again"
3230 print"q          "en;em$" "et" ";es
3240 close2:close3:close15:poke808,237
3250 end
3260 return
3270 print"Sqq        This is not a Geos
 disk"
3280 print"Please rerun program and use a
 Geos disk"
3290 goto 3240
3300 if b>64and b<91 thenb=b+32 :b$=chr$(
b)
3310 if b>192andb<224thenb=b-96 :b$=chr$(
b)
3320 return
3330 print"Sqqqq There is not enough roo
m on this disk"
3340 print"q   Please transfer file to a
nother"
3350 print"q      Geos disk with more sp
ace":goto2980
3360 print"Sqqqqqqq           r error R"

3370 goto3240
3380 get#2,a$:if a$=""then a$=chr$(0)
3390 return
3400 print#15,"u1:2 0 18"ds
3410 gosub3190
3420 print#15,"b-p:2"dp
3430 gosub3190
3440 return
3450 close2:close3:close15:poke808,237
3460 poke53272,(peek(53272)and15)or192:po
```

```
ke648,48
3470 print"S"
3480 open15,8,15:open2,8,2,"geos boot,p,r
":input#15,gg:close2:close15
3490 ifgg=0then3540
3500 print"sqqqqXnInsert Geos Disk in Dri
ve 0."
3510 print"Hit any key to continue..."
3520 geta$:ifa$=""then3520
3530 goto3470
3540 load"geos",8,1
3550 end
3560 end
3570 data76,20,192,76,169,193,76,143,193,
76,198,193,0,0,2,0,0,0,0,0,169,198
3580 data141,133,193,169,0,141,132,193,23
8,15,192,173,18,192,133,251,173,19
3590 data192,133,252,162,0,160,0,177,251,
201,0,208,52,200,208,247,230,252,177
3600 data251,201,0,208,41,200,192,64,208,
245,32,254,193,224,1,240,3,232,208
3610 data222,238,14,192,238,14,192,165,25
1,141,18,192,165,252,141,19,192,173
3620 data15,192,201,13,208,174,133,2,96,1
69,1,141,12,192,173,18,192,133,251
3630 data173,19,192,133,252,162,0,160,0,1
69,63,32,131,193,177,251,32,131,193
3640 data200,192,63,144,246,32,240,193,23
2,224,5,144,231,169,5,32,131,193,160
3650 data0,177,251,32,131,193,200,192,5,1
44,246,32,42,194,162,2,169,255,32
3660 data131,193,169,0,32,131,193,202,208
,243,173,12,192,208,4,169,202,208,2
3670 data169,194,32,131,193,169,0,32,131,
193,173,12,192,240,16,206,12,192,173
3680 data15,192,201,13,208,3,76,12,194,76
,120,192,165,251,141,18,192,165,252
3690 data141,19,192,173,15,192,201,13,208
,4,169,1,133,2,169,1,141,13,192,160
3700 data0,169,40,32,131,193,185,0,92,32,
131,193,200,192,40,144,245,169,168
```

```
3710 data32,131,193,169,191,32,131,193,24
,173,0,193,105,40,141,0,193,173,1
3720 data193,105,0,141,1,193,173,13,192,2
40,20,206,13,192,165,2,208,3,76,248
3730 data192,169,208,32,131,193,169,191,3
2,131,193,169,0,32,131,193,173,132
3740 data193,141,16,192,173,133,193,141,1
7,192,32,204,255,162,3,32,201,255
3750 data169,0,141,98,193,169,198,141,99,
193,173,0,198,32,210,255,238,98,193
3760 data208,3,238,99,193,173,99,193,205,
17,192,208,234,173,98,193,205,16,192
3770 data208,226,32,204,255,96,141,0,198,
238,132,193,208,3,238,133,193,96,172
3780 data14,192,153,0,207,200,138,153,0,2
07,200,240,4,140,14,192,96,169,255
3790 data133,2,32,204,255,96,162,0,169,0,
157,0,207,232,169,255,157,0,207,232
3800 data224,94,144,240,169,0,157,0,207,2
32,208,248,76,218,193,162,2,32,201
3810 data255,160,0,185,0,207,32,210,255,2
00,208,247,32,204,255,96,169,96,141
3820 data19,192,169,92,141,1,193,169,0,14
1,18,192,141,0,193,141,15,192,96,24
3830 data165,251,105,63,133,251,165,252,1
05,0,133,252,96,24,165,251,105,64
3840 data133,251,165,252,105,0,133,252,96
,162,0,160,0,169,255,32,131,193,152
3850 data32,131,193,232,224,5,144,242,169
,141,32,131,193,152,32,131,193,76
3860 data222,192,24,165,251,105,5,133,251
,165,252,105,0,133,252,96
3870 data3,21,191
3880 data255,255,255,128,0,1,145,0,1,138,
117,113,132,85,65,132,119,65,128,0,1
3890 data143,64,1,137,0,1,143,92,241,136,
80,17,136,92,17,128,0,17,255,255,125
3900 data0,0,56,124,0,16,69,239,120,65,9,
64,77,201,120,69,9,8,125,239,120
3910 data131,7,1,0,0,255,255,0,0,80,97,10
5,110,116,32,73,109,97,103,101,32
```

```
3920 data86,49,46,49,0,0,0,0
3930 data83,46,32,71,97,115,116,32,38,32,
82,46,32,82,111,108,108,105,110,115,103
3940 data101,111,80,97,105,110,116,32,32,
32,32,86,49,46,49,0,0,0,0,32,28,63,141
3950 data252,61,32,165,4,32,60,63,169,0,1
41,184,95,169,0,133,17,169,7
3960 data0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
3970 data0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
3980 data0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
```

```
1000 sys700
1010 ;*********************************
1020 ;****   geos code encription   ***
1030 ;**** standard doodle pictures ***
1040 ;****                          ***
1050 ;**** r.rollins - (c) midnight ***
1060 ;*********************************
1100        *= $c000
1110 .opt oo
1120 ;************************
1130 ;routine to change hires
1140 ;screen into geos format
1150 ;with or with out color
1160 ;************************
1170 chkout  = $ffc9
1180 chrout  = $ffd2
1190 clrchn  = $ffcc
1200 donefl  = $0002
1210 databl  = $cf00
1220 ;***********************
1230 dojob   jmp test1
1240 init    jmp datarea
1250 dodata  jmp storets
1260 stdata  jmp ending
1270 ;***********************
1280 timefl  .byte $00
1290 colorfl .byte $00
1300 datapt  .byte $02
1310 linect  .byte $00
1320 storex  .byte $00
1330 storey  .byte $00
1340 pointlo .byte $00
1350 pointhi .byte $00
1360 ;***********************
1370 ; this is where we do the
1380 ; work of converting
1390 ;***********************
1400 ;first we check for a null
1410 ; pair of line in the hires
1420 ;-----------------------
```

```
1430 test1   lda #$c6  ; set up the
1440         sta indata+2 ;storage area
1450         lda #$00
1460         sta indata+1
1470        inc linect  ;what pair of lines we are on
1480         lda pointlo
1490         sta $fb     ;set up counter for the routines
1500         lda pointhi
1510         sta $fc
1520         ldx #$00
1530 test1a  ldy #$00     ;set loop and times done counters
1540 test2   lda ($fb),y  ;check the datafor a null byte
1550         cmp #$00
1560 bne getdata  ;not a null byte so go routines
1570         iny
1575         bne test2
1580        inc $fc      ;raise counter by 256
1590 test3   lda ($fb),y   ;keep going
1600         cmp #$00
1610         bne getdata   ;not a null
1620         iny
1630         cpy #64
1640         bne test3     ;end of line yet
1650         jsr add64     ;add 64 to our counter
1660         cpx #$01      ;first or second line
1670         beq test4     ;second line
1680         inx
1690         bne test1a    ;allways branch
1700 test4   inc datapt    ;raise data block counter by two
1710         inc datapt
1720        lda $fb
1730          sta pointlo
1740          lda $fc      ;reset pointer
1750          sta pointhi  ;for next line
1760          lda linect   ;are we done yet
1770          cmp #13      ;last line
1780          bne test1
1790          sta donefl   ;were done so leave routines
1800          rts
1810 ;*********************
1820 ; this is the convert
```

```
1830 ; program part
1840 ;***********************
1850 getdata  lda #$01
1860          sta timefl
1870          lda pointlo     ;set up counters
1880          sta $fb
1890          lda pointhi
1900          sta $fc
1910 loopset  ldx #$00
1920 loop     ldy #$00
1930          lda #$3f       ;output max no of indivual bytes
1940          jsr indata
1950 loop1    lda ($fb),y    ;get bytes of the hires screen
1960          jsr indata     ;and save to file area
1970          iny
1980          cpy #$3f       ;have all 64 yet
1990          bcc loop1      ;not yet
2000          jsr raisect    ;raise $fb & $fc by 64
2010          inx
2020          cpx #$05        ;have we done this five times
2030          bcc loop       ;not yet so keep doing
2040          lda #$05       ;get last five bytes
2050          jsr indata
2060          ldy #$00
2070 loop2    lda ($fb),y
2080          jsr indata
2090          iny
2100          cpy #$05       ;all five yet
2110          bcc loop2      ;no yet so keep doing
2120          jsr raise5     ;increase $fb & $fc by five
2130          ldx #$02
2140 loop3    lda #$ff       ;pad out the rest of the line
2150          jsr indata
2160          lda #$00
2170          jsr indata
2180          dex
2190          bne loop3
2200          lda timefl
2210          bne loop4      ;second time though
2220          lda #$ca
2230          bne loop5
```

```
2240 loop4    lda #$c2
2250 loop5   jsr indata
2260          lda #$00
2270          jsr indata
2280          lda timefl
2290          beq loop7
2300          dec timefl
2310          lda linect    ;what line we are on
2320          cmp #13       ;last line
2330          bne loop6    ;not last line
2340          jmp lasthlf
2350 loop6    jmp loopset
2360 loop7    lda $fb
2370          sta pointlo   ;save counter fo the enext time
2380          lda $fc
2390          sta pointhi
2400 ;******************************
2410 ; get the color bytes next
2420 ; and stor them into the
2430 ; data sotrage area
2440 ;******************************
2450 docolor lda linect
2460         cmp #13
2470         bne cl1
2480         lda #$01
2490         sta donefl
2500 cl1      lda #$01
2510          sta colorfl
2520 univad   ldy #$00
2530          lda #$28
2540          jsr indata
2550 color    lda $5c00,y
2560          jsr indata
2570          iny
2580          cpy #$28
2590          bcc color
2600          lda #$a8
2610          jsr indata
2620          lda #$bf
2630          jsr indata
2640          clc
```

```
2650          lda color+1
2660          adc #$28
2670          sta color+1
2680          lda color+2
2690          adc #$00
2700          sta color+2
2710          lda colorfl
2720          beq color3
2730          dec colorfl
2740          lda donefl
2750          bne color2
2760          jmp univad
2770 color2   lda #$d0
2780          jsr indata
2790          lda #$bf
2800          jsr indata
2810 color3   lda #$00
2820          jsr indata
2830 ;***************************
2840 ;send data to seq file on
2850 ;disk and return to basic
2860 ;
2870 output  lda indata+1
2880         sta storex   ;save end of data lo byte
2890         lda indata+2
2900         sta storey   ;and end of data hi byte
2910         jsr clrchn
2920         ldx #$03     ;set up disk
2930         jsr chkout   ;to recieve data
2940         lda #$00     ;set up
2950         sta laop+1   ;output loop
2960         lda #$c6
2970         sta laop+2
2980 laop    lda $c600    ;get data from
2990 jsr chrout   ;send to disk
3000         inc laop+1   ;raise loop counters
3010        bne laop2
3020         inc laop+2
3030 laop2   lda laop+2   ;check for
3040         cmp storey   ;end of data
3050         bne laop
```

```
3060        lda laop+1
3070         cmp storex
3080         bne laop     ;keep working
3090 jsr clrchn  ;close drive chanel
3100         rts         ;return to basic
3110 ;***************************
3120 ;subroutine to put data into
3130 ;reserved area
3140 ;
3150 indata  sta $c600
3160        inc indata+1
3170        bne indata2
3180        inc indata+2
3190 indata2 rts
3200 ;****************************
3210 storets ldy datapt
3220         sta databl,y
3230         iny
3240         txa
3250         sta databl,y
3260         iny
3270         beq end
3280         sty datapt
3290         rts
3300 end     lda #$ff
3310         sta donefl
3320         jsr clrchn
3330         rts
3340 ;***************************
3350 datarea  ldx #$00
3360 datare1  lda #$00
3370          sta databl,x
3380          inx
3390          lda #$ff
3400          sta databl,x
3410          inx
3420          cpx #94
3430          bcc datare1
3440 datarea2 lda #$00
3450          sta databl,x
3460          inx
```

```
3470          bne datarea2
3480          jmp set
3490 ;*************************
3500 ending   ldx #$02
3510          jsr chkout
3520          ldy #$00
3530 ending1  lda databl,y
3540          jsr chrout
3550          iny
3560          bne ending1
3570          jsr clrchn
3580          rts
3590 ;************************
3600 set      lda #$60
3610          sta pointhi
3620          lda #$5c
3630          sta color+2
3640          lda #$00
3650          sta pointlo
3660          sta color+1
3670          sta linect
3680          rts
3690 ;************************
3700 raisect  clc
3710          lda $fb
3720          adc #$3f
3730          sta $fb
3740          lda $fc
3750          adc #$00
3760          sta $fc
3770          rts
3780 ;**************************
3790 add64    clc
3800          lda $fb
3810          adc #64
3820          sta $fb
3830          lda $fc
3840          adc #$00
3850          sta $fc
3860          rts
3870 ;************************
```

```
3880 lasthlf  ldx #$00
3890         ldy #$00
3900 last2    lda #$ff
3910          jsr indata
3920          tya
3930          jsr indata
3940          inx
3950          cpx #$05
3960          bcc last2
3970          lda #$8d
3980          jsr indata
3990          tya
4000          jsr indata
4010          jmp loop7
4020 ;*************************
4030 raise5   clc
4040          lda $fb
4050          adc #$05
4060          sta $fb
4070          lda $fc
4080          adc #$00
4090          sta $fc
4100          rts
4110 ;*************************
```

BUILD & UNBUILD

Most telecommunications assumes that transmitted files will be in standard Commodore DOS sequential format. GEOS VLIR files and Commodore REL files cannot be easily be transmitted properly. To get around this, the following two utilities are provided to translate GEOS files so that they can be transmitted with any normal terminal program.

BUILD reads a geoWrite or geoPaint file and creates from that a Commodore DOS SEQ file that contains all the information necessary for UNBUILD to create a new file.

Both programs are straightforward to use. Simply LOAD, RUN, and supply the necessary filenames. It is a good idea, whether BUILDing or UNBUILDing to work with disks that have more than enough room. The only part of the program that might be a bit tricky is the filenames. GEOS stores its filenames as true ASCII while Commodore DOS and most telecommunications software for Commodore computers work with filenames in PETSCII.

When using the BUILD program, enter the filename as it appears in a Commodore BASIC directory listing in lowercase/uppercase mode. BUILD will then find the ASCII filename and create a BUILDED file with a PETSCII filename that telecommunications software will be able to find.

UNBUILD leaves the filename of its geoWrite or geoPaint object file in PETSCII. This will often cause the filename to appear odd within GEOS, but will not affect the way the program runs. If you do not like the appearance of the filename, select the file, then select File/rename. But this is purely cosmetic.

As with all such programs, test run it several times on an unimportant disk. Especially UNBUILD, which write directly to the disk and can corrupt important data if a mistake was made when the program was entered.

The programs require that the source and object (original and created) files be on the same disk, which must be GEOS work disks. You may place BUILDed files on a standard disk, but it must be copied to a GEOS disk before it can be UNBUILDed. Usually it is a good idea to work with disks that have few files on them, as this can lessen the chance of data corruption and speed program execution.

Special characters in the program listing are given in the KEY on page 2-3.

```
10 rem "******************************
20 rem "****   build sequential  *****
30 rem "****     r. rollins      *****
40 rem "****         and         *****
50 rem "****       s. gast       *****
60 rem "****      (c) 1986       *****
70 rem "****    Midnite Press    *****
80 rem "** Midnite Software Gazette **
90 rem "******************************
95 :
100 ifff=1thensys24576
110 forx=49152to 49565
120 read d
130 poke x,d
140 nextx
150 pd$="                ": rem 16 spaces

160 gs$="geos "+chr$(102)+chr$(111)+chr$(
114)+chr$(109)+chr$(97)+chr$(116)
170 printchr$(14):poke808,239
180 poke53280,0:poke53281,15:poke646,11
190 print"Sqqqqq     Insert disk with Geo
s file on it "
200 print"        into disk drive number
8"
210 print"q       Press any key to contin
ue"
220 geta$:if a$="" then 220
230 print"qqq            r Please Wait R"

240 open15,8,15,"i0"
250 gosub 1130
260 open2,8,2,"#"
270 gosub 1130
280 print#15,"u1:2 0 18 0"
290 gosub 1130
300 print#15,"b-p:2 173"
310 gosub 1130
320 b$="":x=0
330 gosub 1310
```

```
340 b$=b$+a$
350 x=x+1
360 if x<11 then 330
370 if b$<>gs$then1200
380 print"Sqqqqqqq"
390 poke198,0:fc$="":fl$=""
400 input"Please enter file name (16 lett
ers only)";fl$
410 if len(fl$)<1or len(fl$)>16thenprint"
Sqqqqqqq               try againqq":goto4
00
420 fc$=right$(pd$,(16-len(fl$)))
430 fc$=fl$+fc$
440 print"Sqqqqqqq        Is this the cor
rect file?"
450 print"qq           r "fc$" R"
460 print"qq                Enterq":print
"   'y' for yes 'n' for no 'q' for quit"
470 poke198,0
480 geta$:if a$=""then480
490 if a$="n"then 380
500 if a$="q" then1180
510 if a$<>"y"then480
520 ac$=""
530 forx=1tolen(fl$):b$=mid$(fl$,x,1):if
b$="*"thenac$=ac$+b$:x=len(fl$):goto550
540 b=asc(b$):gosub1490:ac$=ac$+b$:nextx
550 open3,8,3,"0:"+ac$+",u,r"
560 input#15,en,em$,et$,es$
570 if en>20 then 1410
580 close3
590 print#15,"m-r"chr$(144)chr$(2)
600 get#15,ds$:ifds$=""thends$=chr$(0)
610 ds=asc(ds$)
620 print#15,"m-r"chr$(148)chr$(2)
630 get#15,dc$:ifdc$=""thendc$=chr$(0)
640 dc=asc(dc$)
650 print#15,"u1:"2;0;18;ds
660 print#15,"b-p:"2;dc
670 forx=1to32
680 gosub1310
```

```
690 di$=di$+a$
700 next
710 if asc(mid$(di$,22,1))<>1andasc(mid$(
di$,23,1))<>7 then 1230
720 fs=asc(mid$(di$,29,1))+(asc(mid$(di$,
30,1))*256)
730 print#15,"m-r"chr$(250)chr$(2)
740 get#15,a$:ifa$=""thena$=chr$(0)
750 bf=asc(a$)
760 print#15,"m-r"chr$(252)chr$(2)
770 get#15,a$:ifa$=""thena$=chr$(0)
780 bf=bf+(asc(a$)*256)
790 if fs=>bf then 1520
800 it=asc(mid$(di$,20,1)):is=asc(mid$(di
$,21,1))
810 print#15,"u1:"2;0;it;is
820 gosub 1310
830 print#15,"b-p:2 77"
840 gosub 1310
850 if asc(a$)<>80andasc(a$)<>87 then1330

860 l=len(ac$):ifl>14thenl=14
870 sn$=left$(ac$,l):if mid$(sn$,l,1)="*"
then sn$=left$(sn$,(l-1))
880 for cc=1 to len(sn$)
890 ch$=mid$(sn$,cc,1):ch=asc(ch$)
900 if ch>63 and ch<96 then ch=ch+128
910 if ch>95 and ch<129 then ch=ch-32
920 sm$=sm$+chr$(ch)
930 next
940 open3,8,3,"0:"+sm$+".s,s,w"
950 gosub1130
960 print#3,di$;a$;
970 dt=asc(mid$(di$,2,1)):ds=asc(mid$(di$
,3,1))
980 print#15,"u1:"2;0;dt;ds
990 gosub 1130
1000 poke780,it:poke 781,is
1010 print"sqqqqqqqqqqqqqqqqqqqq
     r WORKING R       1"
1020 sys49152
```

```
1030 er=peek(2):if er<>0 then 1550
1040 print#3
1050 print"Sqqqqqqq          r PROGRAM C
OMPLETE R"
1060 print"q                 Happy to help"
1070 print"qUse unbuild to restore file t
o Geos type"
1080 print"q         Press 'g' to return t
o Geos"
1090 print"q              any other key to qu
it"
1100 getrr$:ifrr$="" then 1100
1110 if rr$<>"g" then 1180
1120 goto 1570
1130 input#15,en,em$,et$,es$
1140 if en<20 then return
1150 print"Sqqqqqqq        r A disk error
 has happened R"
1160 print"   Please rerun program and t
ry again"
1170 print"q              "en;em$" "et$" ";es
$
1180 close2:close3:close15:poke808,237
1190 end
1200 print"Sqq           This is not a Geos
 disk"
1210 print"Please rerun program and use a
 Geos disk"
1220 goto 1180
1230 print"Sqqqqqqqqq          r "fc$" R"

1240 print"   q is not a appclation file
try again"
1250 print"q           Enter 'y' to try aga
in       "
1260 print"q           Any other key to qui
t"
1270 poke198,0
1280 geta$:ifa$=""then1280
1290 if a$<>"y"then1180
1300 close3:goto380
```

```
1310 get#2,a$:if a$=""then a$=chr$(0)
1320 return
1330 print"Sqqqqq        This is a corrup
ted file"
1340 print"q        Program will not work
on it"
1350 print"q    Please rewrite file and t
ry again"
1360 print"q              ]Press 'q' to quit
"
1370 print"q        Any other key to go to
 Geos"
1380 geta$:if a$=""then1380
1390 if a$<>"q"then1570
1400 goto 1180
1410 print"Sqqqqq             r "fc$" R"
1420 print"  q is not on this disk PLEASE
 try again"
1430 print"qq          Enter 'y' to try ag
ain"
1440 print"q          Any other key to qui
t"
1450 poke198,0
1460 geta$:ifa$=""then1460
1470 if a$<>"y"then1180
1480 close2:close3:close15:goto150
1490 if b>192andb<224thenb=b-96 :b$=chr$(
b)
1500 return
1510 printb;asc(b$):return
1520 print"Sqqqq  There is not enough roo
m on this disk"
1530 print"q    Please transfer file to a
nother"
1540 print"q        Geos disk with more sp
ace":goto1360
1550 print"Sqqqqqqq             r error R"

1560 goto1180
1570 close2:close3:close15:poke808,237
1580 poke53272,(peek(53272)and15)or192:po
```

```
ke648,48
1590 print"S"
1600 open15,8,15:open2,8,2,"geos boot,p,r
":input#15,gg:close2:close15
1610 ifgg=0then1660
1620 print"sqqqqXnInsert Geos Disk in Dri
ve 0."
1630 print"Hit any key to continue..."
1640 geta$:ifa$=""then1640
1650 goto1590
1660 ifff=0thenff=1:load"geos boot",8,1
1670 end
1680 data141,154,193,142,155,193,162,2,32
,198,255,160,0,32,207,255,153,0,207
1690 data200,208,247,32,204,255,169,207,1
41,3,193,160,255,32,246,192,169,206
1700 data141,3,193,32,77,193,162,2,32,198
,255,160,0,32,207,255,153,0,206,200
1710 data208,247,32,204,255,160,255,32,24
6,192,160,0,200,208,3,76,212,192,192
1720 data130,208,3,32,16,193,200,240,127,
192,130,208,3,32,16,193,185,0,207
1730 data208,25,200,240,112,192,130,208,3
,76,79,192,185,0,207,240,226,201,255
1740 data208,64,32,127,193,76,82,192,201,
36,176,54,141,154,193,200,240,80,185
1750 data0,207,201,21,176,41,141,155,193,
140,157,193,32,43,193,32,77,193,162
1760 data2,32,198,255,160,0,32,207,255,15
3,0,206,200,208,247,32,204,255,173,0
1770 data206,208,8,172,1,206,208,9,76,238
,192,201,36,176,249,160,255,141,154
1780 data193,173,1,206,141,155,193,32,246
,192,173,154,193,208,198,172,157,193
1790 data76,82,192,169,0,133,2,32,204,255
,162,3,32,201,255,160,3,169,255,32
1800 data210,255,136,208,248,32,204,255,9
6,169,255,133,2,32,204,255,96,140
1810 data156,193,160,255,162,3,32,201,255
,200,185,0,207,32,210,255,204,156
1820 data193,208,244,32,204,255,96,140,15
```

```
7,193,162,3,32,201,255,160,0,169,0
1830 data32,210,255,200,192,3,208,246,32,
204,255,172,157,193,96,169,19,32,210
1840 data255,160,20,169,17,32,210,255,136
,208,248,160,30,169,29,32,210,255
1850 data136,208,248,174,157,193,169,0,32
,205,189,96,162,15,32,201,255,160,0
1860 data185,146,193,32,210,255,200,192,8
,144,245,174,154,193,169,0,32,205
1870 data189,169,32,32,210,255,174,155,19
3,224,21,144,3,76,238,192,169,0,32
1880 data205,189,32,204,255,96,162,3,32,2
01,255,169,64,32,210,255,169,64,32
1890 data210,255,32,204,255,96,85,49,58,3
2,50,32,48,32,0,0,0,0
```

```
10 rem "******************************
20 rem "**** unbuild sequential  *****
30 rem "****     r. rollins      *****
40 rem "****        and          *****
50 rem "****      s. gast        *****
60 rem "****     (c) 1986        *****
70 rem "****   Midnite Press     *****
80 rem "** Midnite Software Gazette **
90 rem "******************************
95 :
100 ifff=1thensys24576
110 forx=49152to 49476
120 read d
130 poke x,d
140 nextx
150 gs$="geos "+chr$(102)+chr$(111)+chr$(
114)+chr$(109)+chr$(97)+chr$(116)
160 pd$="                ":nu$=chr$(0)
170 printchr$(14)
180 poke53280,0:poke53281,15:poke646,11
190 print"Sqqqqq  Insert disk with Seqent
ial file on it "
200 print"        into disk drive number
8"
210 print"q       Press any key to contin
ue"
220 geta$:if a$="" then 220
230 print"qqq            r Please Wait R"

240 open15,8,15,"i0"
250 gosub 1540
260 open2,8,2,"#"
270 gosub 1540
280 print#15,"u1:2 0 18 0"
290 gosub 1540
300 print#15,"b-p:2 173"
310 gosub 1540
320 b$="":x=0
330 gosub 1700
340 b$=b$+a$
```

```
350 x=x+1
360 if x<11 then 330
370 if b$<>gs$ then1610
380 print"Sqqqqqqq"
390 poke198,0:fc$="":fl$=""
400 input"Please enter file name (16 lett
ers only)";fl$
410 if len(fl$)<1or len(fl$)>16thenprint"
Sqqqqqqq              try againqq":goto4
00
420 fc$=right$(pd$,(16-len(fl$)))
430 fc$=fl$+fc$
440 print"Sqqqqqqq       Is this the cor
rect file?"
450 print"qq          r "fc$" R"
460 print"qq               Enterq":print
"   'y' for yes 'n' for no 'q' for quit"
470 poke198,0
480 geta$:if a$=""then480
490 if a$="n"then 380
500 if a$="q" then1590
510 if a$<>"y"then480
520 ac$=""
530 forx=1tolen(fl$):b$=mid$(fl$,x,1):if
b$="*"thenac$=ac$+b$:x=len(fl$):goto570
540 b=asc(b$):ac$=ac$+b$:nextx
550 t=1
560 close2
570 open2,8,2,"0:"+ac$+",s,r"
580 input#15,en,em$,et$,es$
590 if en>20 then 1800
600 gosub 2070
610 forx=1to32
620 gosub1700
630 di$=di$+a$
640 next
650 gosub1700:ft$=a$
660 tf$=mid$(di$,4,16)
670 open3,8,3,"0:"+tf$+",u,r"
680 input#15,en,em$,et$,es$
690 if en<>62then 1960
```

```
700 close3
710 fs=asc(mid$(di$,29,1))+(asc(mid$(di$,
30,1))*256)
720 print#15,"m-r"chr$(250)chr$(2)
730 get#15,a$:ifa$=""thena$=chr$(0)
740 bf=asc(a$)
750 print#15,"m-r"chr$(252)chr$(2)
760 get#15,a$:ifa$=""thena$=chr$(0)
770 bf=bf+(asc(a$)*256)
780 if fs=>bf then 1890
790 open3,8,3,"0:test.s.e.q,s,w"
800 gosub1540
810 sys49152
820 close3
830 open3,8,3,"0:test.s.e.q,s,r"
840 gosub1540
850 gosub2070
860 print#15,"m-r"chr$(144)chr$(2)
870 get#15,ds$:ifds$=""thends$=chr$(0)
880 ds=asc(ds$)
890 print#15,"m-r"chr$(148)chr$(2)
900 get#15,dp$:ifdp$=""thendp$=chr$(0)
910 dp=asc(dp$)
920 close3
930 gosub 2000
940 print#3,nu$;
950 gosub 1940
960 dt$=a$:dt=asc(dt$)
970 gosub 1940
980 sd$=a$:sd=asc(sd$)
990 print#15,"u2:3 0 18"ds
1000 close3
1010 close3
1020 open3,8,3,"0:test.s.e.q,s,w"
1030 gosub1540
1040 gosub2070
1050 sys49155
1060 close3
1070 gosub 2000
1080 print#3,nu$;
1090 gosub1940
```

```
1100 it$=a$:it=asc(it$)
1110 gosub1940
1120 is$=a$:is=asc(is$)
1130 print#15,"u2:3 0 18"ds
1140 close3
1150 di$=mid$(di$,1,1)+dt$+sd$+mid$(di$,4
,16)+it$+is$+mid$(di$,22)
1160 open3,8,3,"0:test.s.e.q,s,w"
1170 gosub1540
1180 gosub2070
1190 sys49158
1200 close3
1210 ck=peek(2):ifck<>0then 1720
1220 fl=peek(49476):if fl<>0 then1350
1230 gosub2000
1240 print#3,nu$;
1250 gosub1940
1260 track=asc(a$)
1270 gosub1940
1280 sectr=asc(a$)
1290 print#15,"u2:3 0 18"ds
1300 close3
1310 poke780,track:poke781,sectr
1320 sys49161
1330 ck=peek(2):ifck<>0then 1720
1340 fl=peek(49476):if fl=0 then1160
1350 gosub 2000
1360 print#3,di$;
1370 print#15,"u2:3 0 18"ds
1380 gosub1540
1390 print#15,"u1:3  0"dt;sd
1400 gosub1540
1410 print#15,"b-p:3 2"
1420 gosub1540
1430 sys49164
1440 print#15,"u2:3 0"dt;sd
1450 gosub 1540
1460 print"Sqqqqqqq           r PROGRAM C
OMPLETE R"
1470 print"q              Happy to help"
```

```
1480 print"q        Press 'g' to return t
o Geos"
1490 print"q          any other key to qu
it"
1500 getrr$:ifrr$="" then 1500
1510 if rr$<>"g" then 1590
1520 goto 2100
1530 end
1540 input#15,en,em$,et$,es$
1550 if en<20 then return
1560 print"Sqqqqqqq        r A disk error
 has happened R"
1570 print"    Please rerun proqram and t
ry again"
1580 print"q          "en;em$" "et$" ";es
$
1590 close2:close3:close15
1600 end
1610 print"Sqq         This is not a Geos
 disk"
1620 print"Please rerun program and use a
 Geos disk"
1630 goto 1590
1640 print"   q is not a appclation file
try again"
1650 print"q         Any other key to qui
t"
1660 poke198,0
1670 geta$:ifa$=""then1670
1680 if a$<>"y"then1590
1690 close3:goto380
1700 get#2,a$:if a$=""then a$=chr$(0)
1710 return
1720 print"Sqqqqq        This is a corrup
ted file"
1730 print"q       Program will not work
on it"
1740 print"q    Please rewrite file and t
ry again"
1750 print"q           ]Press 'q' to quit
"
```

```
1760 print"q       Any other key to go to
 Geos"
1770 geta$:if a$=""then1770
1780 if a$<>"q"then2100
1790 goto 1590
1800 print"Sqqqqq           r "fc$" R"
1810 print"  q is not on this disk PLEASE
 try again"
1820 print"qq        Enter 'y' to try ag
ain"
1830 print"q        Any other key to qui
t"
1840 poke198,0
1850 geta$:ifa$=""then1850
1860 if a$<>"y"then1590
1870 close2:close3:close15:goto160
1880 printb;asc(b$):return
1890 print"Sqqqq  There is not enough roo
m on this disk"
1900 print"q    Please transfer file to a
nother"
1910 print"q       Geos disk with more sp
ace":goto1750
1920 print"Sqqqqqqq           r error R"

1930 goto1590
1940 get#3,a$:if a$="" then a$=chr$(0)
1950 return
1960 print"Sqqqqq]]]]The orignal file is
on this disk"
1970 print"q]]]]Please transfer sequentia
l file "
1980 print"q]]]]]]]]]to another geos disk
"
1990 goto1590
2000 open3,8,3,"#"
2010 gosub1540
2020 print#15,"u1:3 0 18"ds
2030 gosub1540
2040 print#15,"b-p:3"dp
2050 gosub1540
```

```
2060 return
2070 print"sqqqqqqqqqqqqqqqqqqqqq
     r WORKING R      "t
2080 t=t+1
2090 return
2100 close2:close3:close15
2110 poke53272,(peek(53272)and15)or192:po
ke648,48
2120 print"S"
2130 open15,8,15:open2,8,2,"geos boot,p,r
":input#15,gg:close2:close15
2140 ifgg=0then2190
2150 print"sqqqqXnInsert Geos Disk in Dri
ve 0."
2160 print"Hit any key to continue..."
2170 geta$:ifa$=""then2170
2180 goto2120
2190 ifff=0thenff=1:load"geos boot",8,1
2200 end
2210 data76,198,192,76,0,193,76,15,192,76
,180,192,76,45,193,169,0,141,67,193
2220 data133,2,162,2,32,198,255,32,207,25
5,208,16,169,255,141,67,193,32,207
2230 data255,240,88,168,136,208,13,240,12
4,48,94,201,64,240,51,32,207,255,160
2240 data254,140,66,193,32,207,255,153,0,
206,136,208,247,32,204,255,162,3,32
2250 data201,255,172,66,193,185,0,206,32,
210,255,136,208,247,32,204,255,173
2260 data67,193,240,172,169,0,141,68,193,
96,234,234,234,32,207,255,201,64,240
2270 data3,76,172,192,238,65,193,240,36,2
38,65,193,240,31,208,153,32,207,255
2280 data169,130,141,65,193,32,204,255,76
,15,192,32,207,255,48,3,76,172,192
2290 data32,207,255,201,255,208,13,169,25
5,141,68,193,169,0,133,2,32,204,255
2300 data96,169,255,133,2,32,204,255,96,1
72,65,193,153,0,207,200,138,153,0
2310 data207,200,240,221,140,65,193,96,16
```

```
2,2,32,198,255,160,0,32,207,255,153
2320 data0,207,200,32,207,255,153,0,207,2
00,32,207,255,153,0,207,200,208,247
2330 data32,204,255,162,3,32,201,255,160,
2,185,0,207,32,210,255,200,208,247
2340 data32,204,255,169,2,141,65,193,96,1
62,2,32,198,255,32,207,255,32,207
2350 data255,160,2,32,207,255,153,0,206,2
00,208,247,32,204,255,162,3,32,201
2360 data255,160,2,185,0,206,32,210,255,2
00,208,247,32,204,255,96,162,3,32
2370 data201,255,160,2,185,0,207,32,210,2
55,200,208,247,32,204,255,96,0,0,0,0
```

UNSCRATCH

On occasion, you might drop an important file into the Waste Basket. If you realize this immediately, the you may recover the file with the UNSCRATCH utility listed below. This utility will probably not be successful if you have saved any data whatsoever since you scratched the file. If you have saved data since you dropped the file you wish to recover into the Wast Basket, you could make matters worse by attempting to UNSCRATCH it.

If you scratch an important file, the best thing to do is to IMMEDIATELY make a backup of the disk. You may use either the GEOS Disk/copy command, the BACKUP program provided on the GEOS disk, or any other copy utility. Then run UNSCRATCH.

UNSCRATCH will provide you with a list of scratched files. Select the most recently scratched file! If you scratched more than one file, more than one file may be listed, and you may be able to recover more than one file. But if you scratched several files, saved one file, and then scratched yet another file, only that last file scratched may be safely recovered using UNSCRATCH.

This utility should really be used only as a last resort, but as such, it works. Even if you hope not to need it, type it in, test it, and then keep it in a safe place. It may save several hours of work and frustration later.

```
10 rem "******************************
20 rem "**** unscratch geos file *****
30 rem "****                     *****
40 rem "****     r. rollins      *****
50 rem "****                     *****
60 rem "****      (c) 1986       *****
70 rem "****    Midnite Press    *****
80 rem "** Midnite Software Gazette **
90 rem "******************************
95 :
100 ifff=1thensys24576
110 forx=0to2:readm:poke(828+x),m:nextx
120 dim nm$(50),s(50),pp(50),bam%(256)
130 gs$="geos "+chr$(102)+chr$(111)+chr$(
114)+chr$(109)+chr$(97)+chr$(116)
140 printchr$(14)
150 poke53280,0:poke53281,15:poke646,11
160 print"Sqqqqq     Insert disk with scr
atched file"
170 print"        into disk drive number
8"
180 print"q       Press any key to contin
ue"
190 geta$:if a$="" then 190
200 print"qqq            r Please Wait R"

210 open15,8,15,"i0"
220 gosub 1480
230 open2,8,2,"#"
240 gosub 1480
250 print#15,"u1:2 0 18 0"
260 gosub 1480
270 print#15,"b-p:2 173"
280 gosub1480
290 b$="":x=0
300 get#2,a$:if a$=""then a$=chr$(0)
310 b$=b$+a$
320 x=x+1
330 if x<11 then 300
340 if b$<>gs$then1550
```

```
350 t=18:s=1:x=0:z=0
360 print#15,"u1:"2;0;t;s
370 sp=s
380 gosub1580:t=asc(a$)
390 gosub1580:s=asc(a$)
400 if t=0then x=1
410 f=2
420 print#15,"b-p:"2;f
430 gosub 1480
440 gosub 1580
450 if asc(a$)=0then1600
460 f=f+32
470 if x=1 and f>s then 500
480 if f<230 then 420
490 goto360
500 if z=0then 1740
510 rem
520 a=0:b=0:c=0:d=0:e=z
530 a=z:if a>=10then a=10
540 print"Sqqq  The files scrached on thi
s disk areq"
550 print"Number    File Name"
560 print"------    ----------"
570 fory=bto(a-1)
580 if y>50 then  y=a:e=z:goto640
590 if d+1>z then y=a:e=z:goto640
600 print" "c;"       ";
610 printmid$(nm$(y),3,16):
620 c=c+1
630 d=d+1
640 nexty
650 if e>9 then print"qThere are more fil
es. Type 'y' see them."
660 print"q  Select the file number to un
scratch"
670 print"q Select r only R the last file
 scratched"
680 geta$:if a$=""then 680
690 if a$="y"and e>9 then720
700 if a$<"0"orasc(a$)>c+47 then 680
710 goto 770
```

```
720 if y=>50then 510
730 if d=>z then 510
740 a=a+10:b=b+10:c=0:e=e-10
750 if a-z>0 then 540
760 goto 520
770 g=(int((d-1)/10)*10)+val(a$)
780 print"Sqqqqq            ";:print"r  "
;mid$(nm$(g),3,16):print" R  ";
790 print"qq    Is this the file you wan
t?":print"     Press 'y' for yes and 'n'f
or no"
800 geta$:ifa$=""then800
810 if a$="n" then 510
820 if a$<>"y"then800
830 print"Sqqqqqqqq          r unscratche
r working R"
840 print"qqq              r please wait
Rqqq"
850 print#15,"i0"
860 gosub1480
870 close2
880 open2,8,2,"#"
890 gosub 1480
900 print#15,"m-r"chr$(0)chr$(7)chr$(255)

910 fori=0to255:get#15,c$:ifc$=""thenc$=c
hr$(0)
920 bam%(i)=asc(c$)
930 nexti
940 bam%(255)=0
950 t=asc(mid$(nm$(g),19,1)):s=asc(mid$(n
m$(g),20,1))
960 gosub1800
970 t=asc(mid$(nm$(g),1,1)):s=asc(mid$(nm
$(g),2,1))
980 dt=t:ds=s:dp=2
990 gosub1800
1000 :   print#15,"u1:"2;0;dt;ds
1010 :    gosub1480
1020 :    print#15,"b-p:"2;0
1030 :    gosub1480
```

```
1040 :   gosub1580
1050 :    if asc(a$)<>0 then 1940
1060 :    print#15,"b-p:"2;dp
1070 :    gosub1480
1080 :    gosub 1580:t=asc(a$):gosub1580:
s=asc(a$)
1090 :   if t<1ort>35 then 1200
1100 :    print"            r still worki
ng R"(dp/2)"Q"
1110 :    dp=dp+2
1120 :   if dp>255 then 1240
1130 :    gosub1800
1140 :       print#15,"u1:"2;0;t;s
1150 :       gosub 1480
1160 :       gosub 1580:t=asc(a$):gosub15
80:s=asc(a$)
1170 :       if t<1ort>35 then1000
1180 :       gosub 1800
1190 :       goto 1140
1200 :    print"            r still worki
ng R"(dp/2)"Q"
1210 :    dp=dp+2
1220 :   if dp>255 then 1240
1230 :   goto 1000
1240 print#15,"b-p:"2;0
1250 gosub1480
1260 forx=0to255
1270 print#2,chr$(bam%(x));
1280 nextx
1290 print#15,"u2:"2;0;18;0
1300 gosub1480
1310 close2
1320 input#15,en,em$,et$,es$
1330 print#15,"i0"
1340 gosub 1480
1350 open2,8,2,"#"
1360 gosub1480
1370 print#15,"u1:"2;0;18;s(g)
1380 gosub 1480
1390 print#15,"b-p:"2;pp(g)
1400 gosub 1480
```

```
1410 rr$=chr$(131)
1420 print#2,rr$;
1430 gosub 1480
1440 print#15,"u2:"2;0;18;s(g)
1450 gosub1480
1460 input#15,em,em$,et,es
1470 goto 1900
1480 input#15,en,em$,et$,es$
1490 if en<20 then return
1500 print"Sqqqqqqq       r A disk error
 has happened R"
1510 print"   Please rerun program and t
ry again"
1520 print"q         "en;em$" "et$" ";es
$
1530 close2:close15
1540 end
1550 print"Sqq        This is not a Geos
 disk"
1560 print"Please rerun program and use a
 Geos disk"
1570 goto 1530
1580 get#2,a$:if a$=""then a$=chr$(0)
1590 return
1600 b$=""
1610 gosub 1580:b$=b$+a$:gosub1580:b$=b$+
a$:gosub1580:if asc(a$)=0 then 460
1620 b$=b$+a$
1630 f=f+3
1640 forf=fto(f+28)
1650 gosub1580
1660 b$=b$+a$
1670 next f
1680 it=asc(mid$(b$,21,1)):is=asc(mid$(b$
,22,1))
1690 ifit<>1andis<>7then470
1700 nm$(z)=b$
1710 s(z)=sp:pp(z)=f-32
1720 z=z+1
1730 goto470
1740 print"Sqqqqqq       There are no sc
```

```
ratched":print"         files on this di
sk.q"
1750 print"      Rerun program and inser
t":print"   disk with scratched files on
it"
1760 goto 1530
1770 rem ***********************
1780 rem  routine for bam repair
1790 rem ***********************
1800 z=int(s/8):rs%=2Δ(s-(z*8)):poke780,r
s%:sys828:rs%=peek(780)
1810 rt=(z+1)+(t*4):at=t*4:as=at+1
1820 bam%(rt)=bam%(rt)andrs%
1830 gosub 1840:as=as+1:gosub 1850:as=as+
1:gosub1850:bam%(at)=cn
1840 cn=0
1850 forbyte=0to7
1860 bit=bam%(as)and(2Δbyte)
1870 if bit <>0 then cn=cn+1
1880 nextbyte
1890 return
1900 print"Sqqqqqqq        r The unscrat
ch is done R":print"q              Happy
to help
1910 print"qqq    Press any key to retur
n to Geos"
1920 geta$:ifa$=""then1920
1930 goto1960
1940 print"Sqqqqqqqq            r FILE C
ORRUPTED R"
1950 print"qq         unscratcher termin
ated":goto 1530
1960 close2:close15
1970 poke53272,(peek(53272)and15)or192:po
ke648,48
1980 print"S"
1990 open15,8,15:open2,8,2,"geos boot,p,r
":input#15,gg:close2:close15
2000 ifgg=0then2050
2010 print"sqqqXnInsert Geos Disk in Dri
ve 0."
```

```
2020 print"Hit any key to continue..."
2030 geta$:ifa$=""then2030
2040 goto1980
2050 ifff=0thenff=1:load"geos boot",8,1
2060 data 73,255,96
```

EASY LOAD

The following program will alter backup GEOS disks so that you need not always use the original GEOS boot disk. Do not run this program on the original disk. Make a duplicate of the original GEOS disk using the disk copy utility of your choice, then run this utility on the duplicate. Subsequent copies of the duplicate will load as well.

This utility will be effective on any GEOS backup disk or work disk that has GEOS, GEOS BOOT, GEOS KERNAL, and DESKTOP or the disk. GEOS KERNAL must be the third file on the disk for GEOS to work properly. The other files may appear anywhere in the disk directory.

Again, use with caution, and do NOT attempt to run this program on the original GEOS boot disk. Special characters in the program listing follow the KEY given on page 2-3.

```
10 rem "******************************
20 rem "****     easy loader     *****
30 rem "****                     *****
40 rem "****     r. rollins      *****
50 rem "****                     *****
60 rem "****      (c) 1986       *****
70 rem "****    Midnite Press    *****
80 rem "** Midnite Software Gazette **
90 rem "******************************
95 :
100 ifff=1thensys24576
110 printchr$(14)
120 poke53280,0:poke53281,15:poke646,11
130 print"Sqqqqq   Insert disk with Geos
boot file on it "
140 print"        into disk drive number
8"
150 print"q       Press any key to contin
ue"
160 poke198,0
170 geta$:if a$="" then 170
180 print"qqq            r Please Wait R"

190 open15,8,15,"i0"
200 gosub 650
210 open2,8,2,"#"
220 gosub 650
230 open3,8,3,"geos boot"
240 input#15,en,em$,et$,es$
250 if en=62 then 570
260 if en>=20 then 670
270 close3
280 print#15,"m-r"chr$(144)chr$(2)
290 get#15,ds$:ifds$=""thends$=chr$(0)
300 ds=asc(ds$)
310 print#15,"m-r"chr$(148)chr$(2)
320 get#15,dc$:ifdc$=""thendc$=chr$(0)
330 dc=asc(dc$)
340 print#15,"u1:"2;0;18;ds
350 gosub650
```

```
360 print#15,"b-p:"2;dc
370 gosub650
380 gosub 720
390 forx=1to3
400 gosub 740
410 print#15,"u1:2 0"track;sectr
420 gosub650
430 next
440 print#15,"b-p:2 212"
450 gosub650
460 print#2,chr$(162);
470 gosub650
480 print#15,"u2:2 0"track;sectr
490 gosub650
500 print"Sqqqqqqq           r PROGRAM CO
MPLETE R"
510 print"q              Happy to help"
520 print"q        Press 'g' to return to
 Geos"
530 print"q          any other key to qui
t"
540 getrr$:ifrr$="" then 540
550 if rr$<>"g" then 700
560 goto 790
570 print"Sqqqq]]]]Geos boot file is not
 on this disk"
580 print"q]]Please insert the disk with
Geos boot"
590 print"q]]]]]]]]]file on it and try ag
ain"
600 print"qPress 'q' to quit any other to
 try again"
610 poke198,0
620 geta$:if a$="" then 620
630 if a$="q" then 700
640 close2:close3:close15:goto130
650 input#15,en,em$,et$,es$
660 if en<20 then return
670 print"Sqqqqqqq        r A disk error
has happened R"
680 print"    Please rerun program and tr
```

```
y again"
690 print"q            "en;em$" "et$" ";es$

700 close2:close3:close15
710 end
720 get#2,a$:ifa$=""thena$=chr$(0)
730 return
740 gosub 720
750 track=asc(a$)
760 gosub 720
770 sectr=asc(a$)
780 return
790 close2:close3:close15
800 poke53272,(peek(53272)and15)or192:pok
e648,48
810 print"S"
820 open15,8,15:open2,8,2,"geos boot,p,r"
:input#15,gg:close2:close15
830 ifgg=0then880
840 print"sqqqqXnInsert Geos Disk in Driv
e 0."
850 print"Hit any key to continue..."
860 geta$:ifa$=""then860
870 goto810
880 ifff=0thenff=1:load"geos boot",8,1
890 end
```

TELECOMMUNICATIONS

Telecommunicating is discussed more fully elsewhere, in both sections one and in the appendixes. This particular telecommunications program is a variation of the popular Buffterm, which has gone through many hands and has had several names.

The actual terminal program is not that large or hard to type in; the machine language Punter Cl protocol is a bit more difficult. The terminal program can be used without Punter Cl protocol by removing the "if peek(49152)... statement in line 200. This checks for the presence of the Punter protocol in memory, and if it is not there, it loads it. Incidentally, this is the standard Punter Cl protocol, and is available on bulletin boards and in public domain libraries.

If you do type in the Punter protocol, you can save a few minutes typing. Enter line 1210, which has thirty-four zeros, and simply type over the line number with the number of the other lines that have only zeros, namely 1220-1260 and 1400-1530.

The terminal program is easy to use. If you have a 1650 or compatible modem, simply select option 4 to dial the telephone number of a bulletin board or on-line service. When a connection is established, you will go immediately to terminal mode. The <CLR/HOME> key will bring you back to the main menu without disconnecting.

If you are using an intelligent modem such as the Commodore 1670, Hayes, or Avatex, you may communicate with the modem directly from terminal mode. For example, to dial simply enter

```
ATDT1,217,3568056
```

As written, the program does not support 1200 baud communications; the program is written completely in BASIC, with the exception of the Punter Cl protocol, and the Commodore 64 BASIC is not fast enough to handle 1200 baud communciations.

The 0 option, GEOS UTES, is not yet implemented in this version, but all other functions work.

As always, be careful in typing this program in and the first several times you use it. Uploading and downloading are always tricky, and the several versions of Xmodem complicate matters further. This is an excellent program for sending and recieving messages, but if you are going to use a BBS or service that supports Xmodem, you will probably want to download another terminal program from that board.

As written, Xmodem uploads are not supported, you may only download using the Xmodem protocol.

If you are new to telecommunications, the following definitions may help clear things up a bit:

host: the computer that you are communicating with. This could be a BBS or a commercial on-line service such as Compuserve or Delphi.

download: to cause the host computer to send a file to your computer which will save the file to disk.

upload: to cause your computer to send a file to the host computer.

protocol, telecommunication: when two computers talk to each other, they send a stream of ones and zeros back and forth to each other. When they do this, the two computers must agree on what sequence of ones and zeros means what. This telecommunications program is set up for 300 baud, eight data bits, one stop bit, and no parity.

protocol, file transfer: To make life easier, and to insure error free exchange of information, several file transfer protocols have been established. These allow computers to exchange files almost automatically, requiring only that both computers be able to support the chosen protocol, that each computer is informed of the protocol that will be used, and that each computer be given the name of the file to be transferred. This terminal program supports Punter Cl protocol for both upload and download, and Xmodem protocol for download only.

buffer: often times, you will want to capture information being sent by the host computer that is not held in a downloadable file. This is often the case with messages. You may capture this information in a RAM buffer by hitting the <Fl> key. You may later save this buffer to disk or print it out to a printer.

```
10 rem "****************************
20 rem "****  j.o. term/bufterm  *****
30 rem "****                     *****
40 rem "****   jim oldfield, jr  *****
50 rem "****                     *****
60 rem "****       1986          *****
70 rem "****    Midnite Press    *****
80 rem "** Midnite Software Gazette **
90 rem "****************************
95 :
100 rem ---------------------------------
-
110 rem people responsible for this fine
       punter terminal program are:
120 rem         steven punter
130 rem         d. caruana
140 rem         g. donner
150 rem         k. peterson
160 rem         k. hisel
170 rem a. kimball, kermit woodall,
180 rem         j. oldfield jr.
190 rem ---------------------------------
-
200 poke53272,23:clr:print"S":ifpeek(4915
2)<>169thenload"c1.ml",8,1
210 poke53280,0:poke53281,0:poke646,12:po
ke787,0:sys50842:ty$="jao":bs=255
220 sys50951:printchr$(14):xa=fre(x)
230 print"ST-Terminal Mode":print"0-GEOS
UTES":print"1-Seq.file read":print"2-Up L
oad-Punter"
240 print"3-Dn Load-Punter":print"4-1650
comp.dial":print"5-directory"
250 print"6-DUMP Buffer":print"7-CLEAR Bu
ffer":print"8-END Prg":print"X-XMODEM"
260 print"Buffer=F1 ON - F3 OFF"
270 geta$:ifa$="0"then1240
280 ifa$="t"ora$="T"then510
290 ifa$="x"ora$="X"thengosub1010
300 ifa$="d"ora$="D"then400
```

```
310 onval(a$)goto520,870,900,400,320,750,
850,1000,1010:goto270
320 open1,8,0,"$0":get#1,a$,a$
330 get#1,a$,a$:ifa$=""then370
340 get#1,a$,b$:printasc(a$+chr$(0))+asc(
b$+chr$(0))*256;
350 get#1,a$:ifa$=""thenprint:goto330
360 printa$;:goto350
370 close1:close8
380 gosub1310
390 goto230
400 b1=56577:b2=56579:pokeb2,32:pokeb1,0
410 print"SBufterm Autodial Menu":print"S
elect 1. Dial a #    2. Main Menu"
420 print"Make Sure You Set Modem to Data
!"
430 nt=0:print:a$="":inputa$:ifval(a$)<1o
rval(a$)>3thengoto230
440 onval(a$)goto560,230
450 poke781,14:sys59903:goto570
460 forb=1to5:vl=54296:lf=54287:hi=54273:
wf=54276:ad=54277:pokehi,120:pokead,9
470 pokelf,30:pokevl,13:pokewf,21:ford=1t
o160
480 next
490 pokead,0:pokewf,0:pokevl,0:print:poke
631,71
500 poke632,79:poke633,13:poke198,3:print
"S":poke2023,0:print:print:print
510 print"STerminal Mode:":print:goto960
520 input"file name";f$
530 open8,8,8,f$:fori=0to1:get#8,g$:i=st:
printg$;:next:close8,8,8
540 gosub1310:
550 goto230
560 print"Enter Number To Dial";:inputd$:
print" ":poke781,18:sys59903:goto450
570 pokeb1,32:fort=1to800:nextt:d=len(d$)
:dx=(d>8):forx=1tod:n$=mid$(d$,x,1)
580 a=val(n$):ifa=0thena=10
590 gosub600:nextx:forxb=1to(40-d):print"
```

```
 ";:nextxb:goto620
600 forn1=1toa:pokeb1,0:fort=1to70:next:p
okeb1,32
610 fort=1to30:next:nextn1:fort=1to250-d*
300:next:return
620 print"SPress F7 To Stop"
630 print"At the tones you're on":fort=60
0-dx*400to0step-1:geta$
640 ifa$="s"ora$="S"then400
650 ifa$=chr$(136)thenpokeb1,0:goto230:re
m use to be rev s in quotes
660 ifpeek(b1)=47then460
670 nextt:pokeb1,02:forzz=1to1600-dx*400-
7900*(d$="7710723"):next:poke214,14
680 goto450
690 print"File type, (P)rg or (S)eq ?"
700 print"(Hit Return After Up+Dn Loads)"

710 geta$:ifa$=""then710
720 ifa$="p"thenpoke51227,1:i$=i$+",p"+",
r":return
730 ifa$="s"thenpoke51227,2:i$=i$+",s"+",
r":return
740 goto710
750 print"SDUMP BUFFER to Printer or Disk
?"
760 geta$:ifa$=""then760
770 ifa$="p"then800
780 ifa$="d"then810
790 goto760
800 print:print"PRINTER  MODE":sys51034:g
oto230
810 print"DISK MODE, File Name":print
820 i$="":inputi$
830 ifi$=""then230
840 i$=i$+",s,w":open2,8,2,i$:sys52224:cl
ose2:goto230
850 print"SClear Buffer <Y/N>":getg$:ifg$
="y"thensys49152:goto210
860 ifg$="n"then230
870 print"Program Name";:gosub990:ifi$=""
```

```
then460
880 gosub690:open2,8,0,i$:sys49173:sys491
64:ifpeek(512)=1thenclose2:goto460
890 sys49173:poke51224,bs:sys49158:close2
:goto460
900 poke50831,0:print"Save As";:gosub990:
ifi$=""then460
910 sys49173:sys49161:ifpeek(512)=1then46
0
920 t$=","+mid$(ty$,peek(51227),1)+",w":o
pen2,8,2,"0:"+i$+t$
930 forx=1to1300:nextx:print:sys61310
940 sys49173:sys49155:close2:forx=1to1000
:poke668,peek(667)
950 ifpeek(50831)>0thenprint:printpeek(50
831)"Blocks....DONE!":goto460
960 close5:open5,2,0,chr$(6):get#5,a$:pok
e665,75:poke55,128:poke56,24
970 bs=255:ty$="jao":sys49167
980 goto230
990 i$="":inputi$:return
1000 xa=fre(x):clr:print"S":end
1010 rem ------------------------------
1020 rem      xmodem protocols
1030 rem          download
1040 rem   by kermit r. woodall 1985
1050 rem
1060 rem  you must "DIM U%(132)" before
1070 rem      using this routine
1080 rem   modem must be  opened as
1090 rem       device 5 as well
1100 rem ------------------------------
1110 dim u%(132)
1120 c$=chr$(0):input"Filename";fi$:input
"Filetype <S>eq or <P>rg";ft$
1130 open8,8,8,"0:"+fi$+","+ft$+",w":prin
t"DOWNLOADING: ";fi$
1140 forx=1to132:get#5,a$:next:gosub1290
1150 l=0:ch=0:forx=1to132
1160 a$="":get#5,a$:e=peek(663):ifpeek(65
9)=0goto1180
```

```
1170 ife=8thenl=l+1:print"Waiting   ":pri
nt"Q";
1180 ife=0thenl=0:print"Recieving ":print
"Q";:goto1210
1190 poke663,0:ifl>288thenl=0:gosub1290:g
oto1150
1200 goto1160
1210 ifx=1thenifa$=chr$(4)thenprint"DONE
    ":gosub1280:close8:return
1220 u%(x)=asc(a$+c$):ch=ch+u%(x):next x:
ch=ch-u%(132)
1230 if(u%(1)+u%(2)+u%(3)and255)thengosub
1290:print"SOH ERROR ":print"Q";:goto1150

1240 ifu%(132)<>(chand255)thengosub1290:p
rint"Chksum ERR":print"Q";:goto1150
1250 bl=bl+1:print"Block     ¼¼¼¼¼"bl:pri
nt"Q";
1260 ifu%(2)<>blthenbl=bl-1:gosub1280:pri
nt"DUP Block ":print"Q";:goto1150
1270 forx=4to131:print#8,chr$(u%(x));:nex
t x:gosub1280:goto1150
1280 print#5,chr$(6);:ba=0:return
1290 ba=ba+1:ifba=11thenclose8:print"CANC
ELLED ":print#5,chr$(24);:return
1300 print#5,chr$(21);:return
1310 print"press a key"
1320 getzz$:ifzz$=""then1320
1330 return
```

```
10 rem "******************************
20 rem "**** punter c1 protocol  *****
30 rem "****                     *****
40 rem "****                     *****
50 rem "****                     *****
60 rem "****                     *****
70 rem "****    Midnite Press    *****
80 rem "** Midnite Software Gazette **
90 rem "******************************
95 :
100 for x=49152 to 52451:read a:pokex,a:n
ext
110 open2,8,2,"c1.ml,p"
120 poke36,0:poke37,192:poke780,36:poke78
1,228:poke782,204
130 sys65496
140 close2
150 data169,0,44,169,3,44,169,6,44,169,9,
44,169,12,44,169,15,234,76,24,192
160 data76,155,197,133,98,186,142,28,200,
169,48,24,101,98,141,46,192,169,192
170 data105,0,141,47,192,76,63,192,76,81,
192,76,8,196,76,208,195,76,206,196
180 data76,23,197,76,170,197,71,79,79,66,
65,68,65,67,75,83,47,66,83,89,78
190 data141,5,200,169,0,141,0,200,141,1,2
00,141,2,200,169,0,141,6,200,141,7
200 data200,32,123,198,32,242,192,165,150
,208,88,173,1,200,141,0,200,173,2
210 data200,141,1,200,173,0,2,141,2,200,1
69,0,141,4,200,169,1,141,3,200,173
220 data5,200,44,3,200,240,22,172,4,200,1
62,0,189,0,200,217,66,192,208,9,200
230 data232,224,3,208,242,76,190,192,14,3
,200,173,4,200,24,105,3,141,4,200
240 data201,15,208,210,76,95,192,169,255,
141,6,200,141,7,200,76,103,192,238
250 data6,200,208,3,238,7,200,173,7,200,1
3,6,200,240,19,173,6,200,201,7,173
260 data7,200,201,20,144,130,169,1,133,15
```

```
0,76,94,197,169,0,133,150,96,234
270 data152,72,173,155,2,205,156,2,240,22
,172,156,2,177,247,72,238,156,2,169
280 data0,133,150,104,141,0,2,104,168,76,
29,193,169,2,133,150,169,0,141,0,2
290 data104,168,72,169,3,133,186,104,96,1
62,5,32,201,255,162,0,185,66,192,32
300 data210,255,200,232,224,3,208,244,76,
204,255,141,8,200,169,0,141,11,200
310 data169,2,133,98,172,8,200,32,36,193,
169,4,32,81,192,165,150,240,7,198
320 data98,208,243,76,66,193,160,9,32,36,
193,173,13,200,240,5,173,8,200,240
330 data31,173,4,201,141,9,200,141,23,200
,32,127,194,165,150,201,1,240,12
340 data201,2,240,219,201,4,240,4,201,8,2
40,211,96,169,16,32,81,192,165,150
350 data208,201,169,10,141,9,200,160,12,3
2,36,193,169,8,32,81,192,165,150
360 data240,5,206,9,200,208,237,96,169,1,
141,11,200,173,30,200,240,5,160,0
370 data32,36,193,169,11,32,81,192,165,15
0,208,237,169,0,141,30,200,173,4
380 data200,201,0,208,53,173,13,200,208,1
04,238,25,200,208,3,238,26,200,32
390 data100,195,160,5,200,177,100,201,255
,208,22,169,1,141,13,200,173,22,200
400 data73,1,141,22,200,32,100,195,32,68,
195,76,2,194,32,255,194,169,43,44
410 data169,47,32,134,197,160,6,32,36,193
,169,8,32,81,192,165,150,208,234,32
420 data100,195,160,4,177,100,141,9,200,3
2,113,195,162,5,32,201,255,160,0
430 data177,100,32,210,255,200,204,9,200,
208,245,32,204,255,169,0,96,169,42
440 data32,134,197,160,6,32,36,193,169,8,
32,81,192,165,150,208,237,169,10
450 data141,9,200,160,12,32,36,193,169,16
,32,81,192,165,150,240,5,206,9,200
460 data208,237,169,3,141,9,200,160,9,32,
36,193,169,0,32,81,192,206,9,200
```

```
470 data208,241,169,1,96,160,0,169,0,141,
6,200,141,7,200,32,123,198,32,242
480 data192,165,150,208,60,173,0,2,153,0,
201,192,3,176,28,153,0,200,192,2
490 data208,21,173,0,200,201,65,208,14,17
3,1,200,201,67,208,7,173,2,200,201
500 data75,240,11,200,204,9,200,208,194,1
69,1,133,150,96,169,255,141,6,200
510 data141,7,200,76,137,194,238,6,200,20
8,3,238,7,200,173,6,200,13,7,200
520 data240,27,173,6,200,201,6,173,7,200,
201,16,208,158,169,2,133,150,192,0
530 data240,4,169,4,133,150,76,94,197,169
,8,133,150,96,173,22,200,73,1,141
540 data22,200,32,100,195,160,5,173,25,20
0,24,105,1,145,100,200,173,26,200
550 data105,0,145,100,162,2,32,198,255,16
0,7,32,207,255,145,100,200,32,183
560 data255,208,10,204,24,200,208,240,152
,72,76,69,195,152,72,160,5,200,169
570 data255,145,100,76,69,195,72,32,204,2
55,32,155,197,32,121,197,32,155,197
580 data160,4,177,100,141,9,200,32,113,19
5,104,160,4,145,100,32,128,195,96
590 data169,0,133,100,173,22,200,24,105,2
01,133,101,96,169,0,133,100,173,22
600 data200,73,1,24,105,201,133,101,96,16
9,0,141,18,200,141,19,200,141,20
610 data200,141,21,200,160,4,173,18,200,2
4,113,100,141,18,200,144,3,238,19
620 data200,173,20,200,81,100,141,20,200,
173,21,200,42,46,20,200,46,21,200
630 data200,204,9,200,208,218,160,0,173,1
8,200,145,100,200,173,19,200,145
640 data100,200,173,20,200,145,100,200,17
3,21,200,145,100,96,169,0,141,13
650 data200,141,12,200,141,29,200,169,1,1
41,22,200,169,255,141,25,200,141,26
660 data200,32,113,195,160,4,169,7,145,10
0,32,100,195,160,5,169,0,145,100
670 data200,145,100,32,172,193,240,251,16
```

```
9,0,141,0,2,96,169,1,141,25,200,169
680 data0,141,26,200,141,13,200,141,22,20
0,141,5,201,141,6,201,141,12,200
690 data169,7,141,4,201,169,0,32,58,193,1
73,13,200,208,210,32,132,196,208,60
700 data32,204,255,173,9,200,201,7,240,22
,162,2,32,201,255,160,7,185,0,201
710 data32,210,255,200,204,9,200,208,244,
32,204,255,173,6,201,201,255,208,8
720 data169,1,141,13,200,169,42,44,169,43
,32,144,198,32,155,197,169,0,76,40
730 data196,32,204,255,169,47,32,210,255,
173,23,200,141,4,201,169,3,76,40
740 data196,173,0,201,141,14,200,173,1,20
1,141,15,200,173,2,201,141,16,200
750 data173,3,201,141,17,200,32,100,195,1
73,23,200,141,9,200,32,128,195,173
760 data0,201,205,14,200,208,27,173,1,201
,205,15,200,208,19,173,2,201,205,16
770 data200,208,11,173,3,201,205,17,200,2
08,3,169,0,96,169,1,96,169,0,141,25
780 data200,141,26,200,141,13,200,141,22,
200,141,12,200,169,7,24,105,1,141,4
790 data201,169,0,32,58,193,173,13,200,20
8,32,32,132,196,208,16,173,7,201
800 data141,27,200,169,1,141,13,200,169,0
,76,233,196,173,23,200,141,4,201
810 data169,3,76,233,196,169,0,141,0,2,96
,169,0,141,13,200,141,12,200,169,1
820 data141,22,200,141,29,200,169,255,141
,25,200,141,26,200,32,113,195,160,4
830 data169,7,24,105,1,145,100,32,100,195
,160,5,169,255,145,100,200,145,100
840 data160,7,173,27,200,145,100,169,1,14
1,30,200,32,172,193,240,251,169,0
850 data141,0,2,96,238,12,200,173,12,200,
201,3,144,12,169,0,141,12,200,173
860 data11,200,240,7,208,17,173,11,200,24
0,12,162,0,160,0,200,208,253,232
870 data224,120,208,246,96,72,173,25,200,
13,26,200,240,10,173,29,200,208,5
```

```
880 data104,32,210,255,72,104,96,32,126,2
39,173,161,2,201,128,240,246,201
890 data146,240,242,96,32,103,198,32,241,
192,165,150,208,97,173,0,2,41,127
900 data141,0,2,201,13,240,64,201,8,240,3
9,201,34,240,40,201,32,48,73,201,65
910 data144,42,201,91,144,17,201,97,144,3
4,201,123,176,57,56,233,32,141,0,2
920 data76,250,197,24,105,128,76,223,197,
169,20,76,223,197,32,24,199,169,0
930 data133,212,76,18,198,32,24,199,76,18
,198,169,32,141,0,2,32,24,199,169
940 data13,141,0,2,76,250,197,234,234,32,
103,198,32,228,255,240,147,141,0,2
950 data201,19,240,69,201,65,144,13,201,9
1,176,9,24,105,32,141,0,2,76,67,198
960 data173,0,2,201,193,144,10,201,219,17
6,6,56,233,128,141,0,2,201,20,208,5
970 data169,8,141,0,2,201,131,208,5,169,1
6,141,0,2,162,5,32,201,255,173,0,2
980 data32,210,255,32,204,255,76,170,197,
96,169,18,32,210,255,169,32,32,210
990 data255,169,157,32,210,255,169,146,32
,210,255,173,141,2,201,2,208,7,104
1000 data186,236,28,200,208,249,169,1,141
,0,2,96,7,201,42,240,3,238,143,198
1010 data76,210,255,120,169,167,141,20,3,
169,198,141,21,3,88,96,165,203,201,4
1020 data240,4,201,5,208,3,141,19,3,173,1
9,3,201,4,208,46,169,194,141,34,4
1030 data169,47,141,35,4,169,207,141,36,4
,169,206,141,37,4,169,32,141,38,4
1040 data173,134,2,141,34,216,141,35,216,
141,36,216,141,37,216,141,38,216,76
1050 data49,234,201,5,208,249,169,194,141
,34,4,169,47,141,35,4,169,207,141,36
1060 data4,169,198,141,37,4,141,38,4,76,2
12,198,169,0,133,178,133,159,169,25
1070 data133,179,169,156,133,254,96,234,2
34,165,186,201,3,208,7,173,19,3,201
1080 data4,240,7,173,0,2,32,210,255,96,17
```

```
3,0,2,201,20,208,11,164,159,208,2
1090 data198,179,198,159,76,86,199,230,15
9,208,16,230,179,164,179,196,254,208
1100 data8,160,5,140,19,3,76,86,199,164,1
59,145,178,32,210,255,96,169,0,168
1110 data170,32,189,255,169,4,170,160,7,3
2,186,255,32,192,255,162,4,32,201
1120 data255,169,0,133,251,133,253,169,25
,133,252,173,141,2,73,1,208,5,169,0
1130 data76,123,199,173,141,2,201,2,208,1
5,32,174,255,32,231,255,96,0,0,0,0,0
1140 data0,234,234,164,253,177,251,240,3,
76,211,199,164,252,196,179,208,6,164
1150 data253,196,159,240,13,164,253,192,2
55,208,22,230,253,230,252,76,123,199
1160 data169,13,32,210,255,32,210,255,32,
174,255,32,231,255,96,230,253,76,123
1170 data199,32,210,255,76,166,199,234,16
9,0,133,251,133,253,160,21,132,252
1180 data164,253,145,251,196,159,208,7,16
6,252,228,179,208,1,96,192,255,208,2
1190 data230,252,230,253,76,228,199,0,0,8
3,47,66,8,9,8,0,0,0,10,0,0,2,1,60,17
1200 data52,236,60,17,52,236,0,65,0,1,0,1
,246,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
1210 data0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
1220 data0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
1230 data0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
1240 data0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
1250 data0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
1260 data0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
1270 data0,0,0,0,0,0,0,60,17,52,236,202,7
,255,82,32,40,32,89,32,44,32,78,32
1280 data79,82,32,68,69,76,69,84,69,32,41
,32,34,59,81,36,58,139,81,36,178,34
```

```
1290 data68,69,76,69,84,69,34,167,54,50,5
0,48,48,0,16,14,59,242,139,81,36,179
1300 data177,34,89,34,172,52,48,44,52,51,
0,73,13,134,21,84,80,178,83,48,170
1310 data51,172,67,89,170,181,40,67,88,17
3,56,41,58,151,84,80,44,194,40,84,80
1320 data41,175,65,40,67,88,171,56,172,18
1,40,67,88,173,56,41,41,0,82,13,144
1330 data21,137,55,48,48,0,92,13,224,21,1
41,50,48,48,48,0,107,13,234,21,141
1340 data51,48,48,48,58,137,55,48,48,0,11
6,13,88,27,153,34,147,34,0,138,13,98
1350 data27,129,73,178,48,164,54,50,58,15
3,194,40,83,48,170,73,41,59,0,155,13
1360 data108,27,139,73,179,177,54,50,167,
153,34,44,34,59,0,162,13,118,27,130
1370 data73,0,168,13,128,27,128,0,0,0,48,
242,84,178,194,40,54,50,41,172,50,53
1380 data54,170,194,40,54,49,41,170,49,0,
253,13,58,242,133,34,84,79,32,80,82
1390 data73,78,84,69,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
1400 data0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
1410 data0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
1420 data0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
1430 data0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
1440 data0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
1450 data0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
1460 data0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
1470 data0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
1480 data0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
1490 data0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
1500 data0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
1510 data0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
1520 data0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
1530 data0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
1540 data0,0,0,0,0,0,0,0,0,169,0,133,251,
133,253,169,25,133,252,162,2,32,201
1550 data255,173,141,2,201,2,208,7,32,174
,255,32,204,255,96,164,253,177,251
1560 data240,3,76,79,204,164,252,196,179,
208,6,164,253,196,159,240,13,164,253
1570 data192,255,208,18,230,253,230,252,7
6,15,204,169,13,32,210,255,32,210
1580 data255,76,22,204,230,253,76,15,204,
32,210,255,76,38,204,234,0,0,0,0,0,0
1590 data0,0,0,0,169,36,133,251,169,251,1
33,187,169,0,133,188,169,1,133,183
1600 data169,8,133,186,169,96,133,185,32,
213,243,165,186,32,180,255,165,185
1610 data32,150,255,169,0,133,144,160,3,1
32,251,32,165,255,133,252,164,144
1620 data208,66,32,165,255,164,144,208,59
,164,251,136,208,233,166,252,32,205
1630 data189,169,32,32,210,255,32,165,255
,166,144,208,37,170,240,6,32,210,255
1640 data76,172,204,169,13,32,210,255,173
,141,2,73,1,208,5,169,0,76,193,204
1650 data173,141,2,201,2,240,4,160,2,208,
179,32,66,246,96,0,0,0,0,0,0,0,0
```

Inside GEOS

GEOS is more than just another program for the Commodore 64. It is a whole new operating system that completely takes over the computer and disk drive. The Commodore 64 Kernal is seldom used, the BASIC 2.0 ROM is almost never mapped into memory, and much of zero page is altered. When GEOS is running, the computer can hardly be called a Commodore 64.

As an operating system, it is sophisticated and well developed. It is, in fact, more complete than the standard Commodore 64 operating system in that it also controls the disk drive. GEOS has its own routines for reading the keyboard, for placing text and graphics on the screen, for communicating with the disk drive, and for managing memory.

A programmer must well understand all these of the GEOS operating system before he can take full advantage of its power. It is not yet clear whether GEOS will remain a closed operating system, or if Berkeley Softworks will give complete documentation for it. Until Berkeley Softworks releases full documentation for the GEOS operating system, programmers who want to develop utilities and programs for GEOS must rely on third party resources and each other for information.

The information on the following pages is a first step toward understanding how GEOS works. It is not complete. The information has been discovered by a team of programmers who have torn the program apart byte by byte and nearly bit by bit.

As such, it is a working document. The information on the following pages can be used as reference in further searches. It is useful as a starting point and a reference from which you can further discover the inner workings of GEOS. We encourage you to share further information that you may discover - commercially or for free, but in any case openly - that more programmers and developers can support GEOS with application programs, accessories, and utilities.

All the information in this book pertains to GEOS V1.2 as shipped during the summer and fall of 1986.

ASCII vs PETSCII

A big area of confusion exists around the differences that exist between ASCII and PETSCII. ASCII (American Standard Code for Information Interchange) is the standard code for microcomputers, and is used by MS DOS machines, CP/M machines, and others.

True ASCII provides one hundred and twenty-eight codes, numbered 0-127, that represent characters. Control characters all have values from 0 to 31, punctuation and numerals have values from 32 through 63, upper case characters have values from 64 through 95, and lower case characters have values from 96 through 127.

PETSCII is a variation of the ASCII code used by the Commodore PET and VIC series of computers. It was first used on the Commodore PET 2001, from which it got its name. PETSCII is also used on the VIC series of computers. (The VIC series includes the VIC 20, C64, and C128. The series draws its name from the Video Interface Chip, or VIC, chip, which is present in some form in these three computers.) PETSCII provides two hundred and fifty six codes, numbered 0-255, that represent characters. It is nearly identical with ASCII for values through 63. At values 64-95, however, PETSCII has lower case characters; from 96 through 191 it has graphics and reversed characters; from 192 through 218 it has upper case characters; and from 219 through 255 it has more graphics characters.

To further complicate matters, the Commodore 64 has two character sets which can be used outside of GEOS. The default character set shows a lower case "a" as an upper case "A," and an upper case "A" as a graphic character that looks like a spade. Thus this is known as the uppercase/graphics character set, uppercase/graphics mode, or local cursor up mode. You may toggle into lowercase/uppercase, or local cursor down, mode by pressing the Commodore logo key (<logo>) and either of the <SHIFT> keys at the same time. A lower case "a" will then appear as an "a," and a <SHIFT>ed "a" will appear as "A."

What appears on your screen as an "a" in lowercase/uppercase mode, or as an "A" in uppercase/graphics mode is represented in the computer as the value 65. You can check this by using the CHR$() function or the ASC() function. For example, in uppercase/graphics mode, enter the following commands:

```
   PRINT CHR$(65)
and
   PRINT ASC("A")
```

Now toggle between the uppercase/graphics and lowercase/uppercase using the <logo><SHIFT> combination. Now do the same with a <SHIFT>ed "A" and the value 193.

To really confuse matters PRINT CHR$(97) in lowercase/uppercase mode. Another upper case "A!" But this is not really an "A." This is a graphics character in Character Set 2 that just looks like a real "A."

GEOS uses true ASCII. This will seldom be a problem, except when you use utilities such as the sequential file BUILD and UNBUILD programs in section two. To show the trouble, we will use the directory from BASIC.

First, load the directory of a GEOS disk from BASIC, or use a DOS wedge to list it.

```
   LOAD"$",8 <RETURN>
      wait for <READY> prompt
   LIST <RETURN>
or
   @$ <RETURN>
      using the DOS Wedge.
```

If you are in uppercase/graphics mode, the listing of the directory will have some upper case characters and many graphics. If you are in lowercase/uppercase mode, the directory will look something like:

```
0   "geos v1.2         " df 2a
1      "geos"              prg<
6      "geos boot"         prg<
85     "geos kernal"       usr<
119    "geopaint"          usr<
88     "geowrite"          usr<
17     "NOTE PAD"          usr
16     "backup"            prg<
22     "PREFERENCE MGR"    usr
23     "PHOTO MANAGER"     usr
20     "TEXT MANAGER"      usr
             .
             .
             .
```

Now we are going to use a little known trick of the Commodore DOS. We can ask the disk drive if a certain file is on the disk. For example, in lowercase/uppercase mode:

```
    load ”$:geos”,8 <RETURN>
        wait for <READY> prompt, and
    list <RETURN>
or
    @$:geos <RETURN>
```

The computer will display a directory with only the ”geos” file. You can do the same with ”geos boot,” ”geos kernal,” ”geopaint,” ”geowrite,” and ”backup.” Now try the same thing using upper case letters on a file like ”NOTE PAD” or ”PREFERENCE MGR.” You will get the first line of the directory and the number of blocks free, but no file names.

This is because when you ask the disk drive to look for a file with the name "NOTE PAD," you are telling it to look for a file that has the codes

[206,207,212,197,32,208,193,196],

but the name of the file on disk has the codes

[110,111,116,101,32,112,97,100].

In lowercase/uppercase mode, the codes look the same. The BUILD program in section 2 compensates for this with the little routine at 1490. If this routine were not used, you would not be able to build a sequential file from a GEOS file unless you had used all upper case letters within GEOS for the file name. The trouble does not end there. The routine at 1490 just lets you find the file on disk. The BUILD program must give a name to the sequential file that it creates. If it used the same codes as the original name, your telecommunications program might never be able to find the program you want to send. The routine in lines 880-930 build a name that a terminal program can use for the new file.

GEOS is able to handle PETSCII for filenames, so this will not cause a problem when you UNBUILD the file back into GEOS format.

FILE STRUCTURE

GEOS is a disk based operating system. Very few programs which run under GEOS are wholly resident when running. Rather, idle code in the computer is replaced by code from the disk as various features are called. Most GEOS programs are also very large and allow little space for the text and graphics data being manipulated. GEOS documents are also only partially loaded into memory, then saved to disk before the next part of the document is loaded.

The most important aspect of GEOS to understand is the structure of files on disk. If this is not understood, all the understanding of how GEOS works within the computer will be of little value. GEOS does provide user callable disk access routines, and when you write GEOS programs you will want to use these. But first you must understand the effects.

GEOS will work with Commodore 1541 and compatible disk drives. It will not work with a dual disk drive such as the MSD SD 2 or the Commodore 4040. At the actual disk surface level, GEOS uses the standard disk controller built into the drive. A GEOS disk stores the bits on disk in standard Commodore GCR code, using the normal total of 683 blocks of 256 bytes each. At this point, differences begin to appear.

Track 18, block 0 contains the standard disk name, id, format, DOS version, and Block Allocation Map (BAM). It also contains special information used by GEOS. Directory entries begin on 18/1 and contain all the normal information, plus another track and sector pointer, a GEOS file type identifier, and date and time stamps. The full disk file structure is explained below.

WORDS

Before we talk too much about the file structure, let us get a few terms straight. Communication can break down very quickly when we are using two number systems and counting may begin at either one or zero.

disk: 5 1/4" floppy. Data is recorded on the top side of the disk as you insert it into a drive.
drive: Commodore 1541 or compatible.
DOS: the standard 1541 operating system.
controller: the electronic circuit that tells the motor to spin the disk, moves and activates the read/write head.
GEOS: a computer and disk operating system that replaces the DOS.
track: a circular path on the disk along which information is stored. The disk has 35 tracks. Track 35 is on the inside of the disk, near the center; track 1 is on the outside of the disk.
block: 256 bytes of information stored on a track. A track may have 17 to 21 blocks. Blocks are numbered beginning at zero (0), the bytes on a block are counted in decimal (base 10) from 0 to 255. The first two bytes are used by the DOS and by GEOS as track/block pointers. The first byte (byte zero) points to the next track of the file, the second byte (byte one) points to the block of that track.
sector: another term for block.
#/#: a specific block will often be referred to by this notation. The first number designates the track, the second number designates the block on that track. For example, the disk name, BAM, and other information is stored on 18/0 (track 18, block 0)
pointer: two bytes within a block that indicate where on the disk the file begins, continues, or more information about the file may be found. The first byte of the pointer indicates the track, the second byte of the pointer indicates the block.

BAM: block allocation map. The BAM consists of 139 bytes in 18/0. Each block on a disk corresponds to one bit of one byte of the BAM. If a bit for a particular block is set, the drive will not write to that block on the disk.

directory: contains information regarding the disk name, the name and location of files on the disk, and the BAM. The directory occupies all of track 18, although only a small part of it may be used at any one time. The directory may hold up to 144 filenames.

auxiliary directory: a single block which GEOS uses to store directory information about files that appear on the border of the deskTop screen.

directory entry: 32 bytes of data which contain file type, file name, and other information.

directory information: the information about a file that is contained within a directory entry.

hexadecimal: base 16. One digit place may have a decimal value of 0 to 15. Digits representing decimal values of zero through nine are the same as decimal. A decimal 10 is represented by the letter "A," decimal 11 = "B,"...15 = "F." Hexadecimal is especially convenient for representing decimal numbers between 0 and 255. Hexadecimal numbers will be preceded by a dollar sign (i.e. $ff = 255).

decimal: base ten. Our standard counting system. Base 10 numbers may be preceded by the plus sign (+).

binary: base two. Each digit of a binary number may have one of two values, 0 or 1. Binary numbers usually have 8 digits, or bits, that represent the value of one byte as found on disk or the contents of one memory location in the computer. The eight digits are numbered zero to seven from the right and each digit may be referred to as a bit (e.g. the binary number %1000 1001 has bits set in positions zero, three, and seven). The value of each position is equal to 2 to the power of the position. The above binary number has the decimal value equal to +137, or in hexadecimal, $89. Binary numbers are preceded by a percent sign, and are often broken into two parts.

The above list of definitions is not exhaustive, but if you become familiar with them you will have sufficient information to go on.

The biggest problem is with counting. As noted above, a block on disk has two hundred and fifty-six bytes numbered 0 through 255. Byte positions will normally be given in decimal unless otherwise noted.

Directory entries each have thirty-two bytes numbered from 0 through 31. Byte positions of directory entries will be given in decimal counting from the byte 0 of the directory entry, not of the disk block.

Each disk has thirty-five tracks numbered from 1 to 35. Track numbers will normally be given in decimal unless otherwise noted. The 1541 disk drive is capable of reading and writing beyond track 35 with machine language programming.

A track has seventeen to twenty-one blocks, depending on its location on the disk, numbered 0 to 17 (18, 19, 20, or 21).
Track thirty-five has seventeen blocks numbered from 0 to 16; track one has twenty-one blocks numbered from 0 to 20. Track numbers will be given in decimal unless otherwise noted.

In order to describe the disk file structure, we have included maps of individual disk blocks. The maps include byte positions from 0 to 255, the value of the byte in each location, and a representation of how that character appears using many sector editors.

The maps have eight blocks of three lines each. Each block contains the information for thirty-two bytes. The top row of each block gives the decimal byte position; the second row gives the hexadecimal value of the byte in that position; the third row gives a character representation of that byte.

Byte positions above one hundred are given in two digits. Positions 100 and 200 are marked with a digit above the top row of the block.

Bytes are represented as nearly as possible by the screen code corresponding to the value of the byte, with reversed characters preceded by a hyphen. For example, a CHR$(1) is represented by "a," CHR$(65), $41, is represented by "A," and CHR$(97), $61, is represented by "-a." A shifted space, CHR$(160), $a0, is represented by an "ss."

Some graphics characters cannot be well represented. The boxes for these characters will either have a character similar to the graphic, or be left blank.

Some sections of data within a block will be noted with text below the block, or with text within the block. The geoWrite data block has text imbedded to highlight the features.

TRACK 18, BLOCK 0

Disk Name, BAM, GEOS Identification

Track 18, block 0 is the first block of the disk directory. It identifies the disk and the particular version of Commodore DOS by which it was formatted.

Positions	Contain
0-1	pointer to first track and sector of file directory
2	CHR$(65) indicating 1541 format
3	not used
4-143	Block Allocation Map (BAM)
144-159	Disk name padded to 16 characters with shifted spaces [CHR$(160)]
160-161	shifted spaces
162-163	disk id
164	shifted space
165-166	DOS version (2A)
177-170	shifted spaces
171-172	pointer to GEOS auxiliary directory block. This is usually 19/8, if available.
173-188	GEOS disk identifier, "GEOS format V1.0"
189-255	nulls ($00) unused by DOS or GEOS

The only difference in 18/0 between a DOS disk and a GEOS disk is at locations 171 through 188. These bytes must be present on a GEOS disk. If these bytes are not present on a disk when you select it from deskTop, GEOS will inform you that ”[disk name] is not a GEOS disk. Would you like it converted?” Selecting ”yes” will cause GEOS to write this information to the disk and to create an auxiliary directory block.

NOTE: Do NOT place a GEOS disk in a Commodore 4040 disk drive and close the door. The 4040 is a very intelligent machine; if it gets bored, it will rewrite the BAM with standard information - replacing the GEOS disk identifier with nulls.

Track 18 block 0 map

```
| 0| 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|27|28|29|30|31|
|12| 1|41|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|
| r| a| A| @| @| @| @| @| @| @| @| @| @| @| @| @| @| @| @| @| @| @| @| @| @| @| @| @| @| @| @| @|


|32|33|34|35|36|37|38|39|40|41|42|43|44|45|46|47|48|49|50|51|52|53|54|55|56|57|58|59|60|61|62|63|
|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|
| @| @| @| @| @| @| @| @| @| @| @| @| @| @| @| @| @| @| @| @| @| @| @| @| @| @| @| @| @| @| @| @|


|64|65|66|67|68|69|70|71|72|73|74|75|76|77|78|79|80|81|82|83|84|85|86|87|88|89|90|91|92|93|94|95|
|00|00|00|00|00|00|00|00|0e|bc|fd|05|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|
| @| @| @| @| @| @| @| @| n| <| ]| e| @| @| @| @| @| @| @| @| @| @| @| @| @| @| @| @| @| @| @| @|

             1
|96|97|98|99|00|01|02|03|04|05|06|07|08|09|10|11|12|13|14|15|16|17|18|19|20|21|22|23|24|25|26|27|
|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|
| @| @| @| @| @| @| @| @| @| @| @| @| @| @| @| @| @| @| @| @| @| @| @| @| @| @| @| @| @| @| @| @|


|28|29|30|31|32|33|34|35|36|37|38|39|40|41|42|43|44|45|46|47|48|49|50|51|52|53|54|55|56|57|58|59|
|00|00|00|00|00|00|00|00|00|00|00|00|01|00|08|00|47|45|4f|53|20|56|31|2e|32|a0|  |  |  |  |  |  |
| @| @| @| @| @| @| @| @| @| @| @| @| a| @| h| @| G| E| O| S|  |V | 1| .| 2|ss|ss|ss|ss|ss|ss|ss|


|60|61|62|63|64|65|66|67|68|69|70|71|72|73|74|75|76|77|78|79|80|81|82|83|84|85|86|87|88|89|90|91|
|a0|a0|44|46|a0|32|41|a0|a0|a0|a0|13|08|47|45|4f|53|20|66|6f|72|6d|61|74|20|56|31|2e|30|00|00|00|
|ss|ss| d| f|ss| 2| A|ss|ss|ss|ss| s| h| G| E| O| S|  |-f|-o|-r|-m|-a|-t|  | V| 1| .| 0| @| @| @|
                                  1 --- 2 ----------------------------------------------
                         2
|92|93|94|95|96|97|98|99|00|01|02|03|04|05|06|07|08|09|10|11|12|13|14|15|16|17|18|19|20|21|22|23|
|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|
| @| @| @| @| @| @| @| @| @| @| @| @| @| @| @| @| @| @| @| @| @| @| @| @| @| @| @| @| @| @| @| @|


|24|25|26|27|28|29|30|31|32|33|34|35|36|37|38|39|40|41|42|43|44|45|46|47|48|49|50|51|52|53|54|55|
|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|
| @| @| @| @| @| @| @| @| @| @| @| @| @| @| @| @| @| @| @| @| @| @| @| @| @| @| @| @| @| @| @| @|
```

TRACK 18, BLOCKS 1 through 18 and TRACK 19, BLOCK 8
Directory Blocks

Directory blocks contain information about DOS file type, location, name, location of the GEOS file information block, GEOS file type, date and time stamp, and size.

Directory blocks contain the directory information for up to eight files. The directory information for the first file occupies positions 0 through 31, the information for the second file occupies positions 32 through 63, etc.

The first two bytes, 0 and 1, of the first directory entry are used to point to the next directory block, if there is one, or to flag the current directory block as the last directory block and indicate the last valid position within the current block. The first two bytes of all other directory entries within a block will be nulls.

Byte 2 indicates the DOS file type. All GEOS files with the exception of "GEOS" and "BACKUP" are USR files. Commodore BASIC and machine language programs are PRG files, certain data files are SEQ files, and standard DOS relative files are REL files. A file type is determined by the lowest three bits of the file type byte. When a file is properly closed bit seven is set, or, if locked, bits seven and six are set. A locked file is indicated in a BASIC directory by a less-than (<) sign following the file type.

If bit seven is not set, this indicates a splat file. A splat file occurs when a file is improperly closed. I have never seen GEOS create a splat file, but it can happen from other programs or from BASIC. If a splat file occurs on a GEOS disk, you may be in deep trouble. Use GEOS to copy all important data files, then reformat the disk. A splat file is indicated by an asterisk (*) preceding the file type.

File type	normal	locked	splat
SEQ	$81	$c1	$01
PRG	$82	$c2	$02
USR	$83	$c3	$03
REL	$84	$c4	$04

All GEOS files are USR files, and several of these are locked. GEOS has several file types of its own, and uses two bytes (23-24) of the directory entry as GEOS file type identifiers. These types are also indicated by bytes in the file information block (see below), but with the order of the bytes reversed. A value of 1 in position 23 indicates that it is a VLIR file - the pointer in positions 3 and 4 point to a side sector. These file types are listed in the deskTop code at $39a5 to $3a27.

23	24	GEOS file type
00	01	BASIC program (BACKUP)
00	02	assembly prg (machine language) (GEOS Boot)
00	03	data file? (unused, present in deskTop list)
00	04	System file (Scraps, Preferences)
01	04	System prgs* (Kernal, deskTop, and Notes)
00	05	desk accessory (Notepad, Pref.Mgr., Calc.)
01	06	application (geoWrite or geoPaint)
01	07	application data (geoWrite, geoPaint, Albums)
01	08	font file (Roma, Dwinelle)
00	09	printer driver
00	0a	input driver

* not present in deskTop table at $39a5

The table below shows the meaning of each byte within a directory entry. The positions given are for the first directory entry, but are valid for subsequent entries by adding an offset equal to (entry # - 1) * (32) + (position). For example, the DOS file type for the third directory entry is at position (3-1)*32+2 = 66.

positions	contents
0-1	pointer. used only in the first directory entry, as above
2	DOS file type. See Standard Commodore Files, below
3-4	pointer to actual file data (sequential type files) or to side sector (VLIR files)
5-20	file name. GEOS files use ASCII rather than PETSCII. File names are padded with shifted spaces [CHR$(160)] to position 20
21-22	pointer to GEOS information block
23-24	GEOS file type. See GEOS files, below.
25-27	date stamp in yy/mm/dd format
28-29	time stamp in hh/mm format
30-31	file size, in blocks, including side sectors and information blocks.
32-63	second directory entry, with the same pattern as above.
64-95	third directory entry, etc.

BLOCK MAP 18/1

```
¦ 0¦ 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¦27¦28¦29¦30¦31¦
¦12¦09¦c2¦01¦08¦47¦45¦4f¦53¦a0¦a0¦a0¦a0¦a0¦a0¦a0¦a0¦a0¦a0¦a0¦a0¦00¦00¦00¦00¦00¦00¦00¦00¦00¦01¦00¦
¦ r¦ i¦-B¦ a¦ h¦ G¦ E¦ O¦ S¦ss¦ss¦ss¦ss¦ss¦ss¦ss¦ss¦ss¦ss¦ss¦ss¦ @¦ @¦ @¦ @¦ @¦ @¦ @¦ @¦ @¦ a¦ @¦

¦32¦33¦34¦35¦36¦37¦38¦39¦40¦41¦42¦43¦44¦45¦46¦47¦48¦49¦50¦51¦52¦53¦54¦55¦56¦57¦58¦59¦60¦61¦62¦63¦
¦00¦00¦c2¦01¦11¦47¦45¦4f¦53¦20¦42¦4f¦4f¦54¦a0¦a0¦a0¦a0¦a0¦a0¦a0¦01¦09¦00¦02¦56¦04¦07¦0c¦00¦06¦00¦
¦ @¦ @¦-B¦ a¦ q¦ G¦ E¦ O¦ S¦  ¦ B¦ O¦ O¦ T¦ss¦ss¦ss¦ss¦ss¦ss¦ss¦ a¦ i¦ @¦ b¦ V¦ d¦ g¦ l¦ @¦ f¦ @¦

¦64¦65¦66¦67¦68¦69¦70¦71¦72¦73¦74¦75¦76¦77¦78¦79¦80¦81¦82¦83¦84¦85¦86¦87¦88¦89¦90¦91¦92¦93¦94¦95¦
¦00¦00¦c3¦01¦18¦47¦45¦4f¦53¦20¦4b¦45¦52¦4e¦41¦4c¦a0¦a0¦a0¦a0¦a0¦01¦0a¦01¦04¦56¦04¦07¦0c¦00¦55¦00¦
¦ @¦ @¦-C¦ a¦ r¦ G¦ E¦ O¦ S¦  ¦ K¦ E¦ R¦ N¦ A¦ L¦ss¦ss¦ss¦ss¦ss¦ a¦ j¦ a¦ d¦ V¦ d¦ g¦ l¦ @¦ U¦ @¦

            1
¦96¦97¦98¦99¦00¦01¦02¦03¦04¦05¦06¦07¦08¦09¦10¦11¦12¦13¦14¦15¦16¦17¦18¦19¦20¦21¦22¦23¦24¦25¦26¦27¦
¦00¦00¦c3¦05¦08¦44¦45¦53¦4b¦20¦54¦4f¦50¦a0¦a0¦a0¦a0¦a0¦a0¦a0¦a0¦05¦00¦01¦04¦56¦04¦07¦0c¦00¦48¦00¦
¦@ ¦@ ¦-C¦ e¦ h¦ D¦ E¦ S¦ K¦  ¦T ¦ O¦ P¦ss¦ss¦ss¦ss¦ss¦ss¦ss¦ss¦ e¦ @¦ a¦ d¦ V¦ d¦ h¦ l¦ @¦ H¦ @¦

¦28¦29¦30¦31¦32¦33¦34¦35¦36¦37¦38¦39¦40¦41¦42¦43¦44¦45¦46¦47¦48¦49¦50¦51¦52¦53¦54¦55¦56¦57¦58¦59¦
¦00¦00¦c3¦08¦13¦47¦45¦4f¦50¦41¦49¦4e¦54¦a0¦a0¦a0¦a0¦a0¦a0¦a0¦a0¦08¦0b¦01¦06¦56¦04¦07¦0c¦00¦77¦00¦
¦ @¦ @¦-C¦ h¦ s¦ G¦ E¦ O¦ P¦ A¦ I¦ N¦ T¦ss¦ss¦ss¦ss¦ss¦ss¦ss¦ss¦ h¦ k¦ a¦ f¦ V¦ d¦ g¦ l¦ @¦--¦ @¦

¦60¦61¦62¦63¦64¦65¦66¦67¦68¦69¦70¦71¦72¦73¦74¦75¦76¦77¦78¦79¦80¦81¦82¦83¦84¦85¦86¦87¦88¦89¦90¦91¦
¦00¦00¦c3¦09¦0e¦47¦45¦4f¦57¦52¦49¦54¦45¦a0¦a0¦a0¦a0¦a0¦a0¦a0¦a0¦08¦06¦01¦06¦56¦04¦07¦0c¦00¦58¦00¦
¦ @¦ @¦-C¦ i¦ n¦ G¦ E¦ O¦ W¦ R¦ I¦ T¦ E¦ss¦ss¦ss¦ss¦ss¦ss¦ss¦ss¦ h¦ f¦ a¦ f¦ V¦ d¦ g¦ l¦ @¦--¦ @¦

                        2
¦92¦93¦94¦95¦96¦97¦98¦99¦00¦01¦02¦03¦04¦05¦06¦07¦08¦09¦10¦11¦12¦13¦14¦15¦16¦17¦18¦19¦20¦21¦22¦23¦
¦00¦00¦c2¦13¦00¦42¦41¦43¦4b¦55¦50¦a0¦a0¦a0¦a0¦a0¦a0¦a0¦a0¦a0¦a0¦13¦0b¦00¦01¦56¦06¦13¦0c¦0C¦11¦00¦
¦ @¦ @¦-B¦ s¦ @¦ B¦ A¦ C¦ K¦ U¦ P¦ss¦ss¦ss¦ss¦ss¦ss¦ss¦ss¦ss¦ss¦ s¦ k¦ @¦ a¦ V¦ f¦ s¦ l¦ @¦ q¦ @¦

¦24¦25¦26¦27¦28¦29¦30¦31¦32¦33¦34¦35¦36¦37¦38¦39¦40¦41¦42¦43¦44¦45¦46¦47¦48¦49¦50¦51¦52¦53¦54¦55¦
¦00¦00¦00¦00¦00¦00¦00¦00¦00¦00¦00¦00¦00¦00¦00¦00¦00¦00¦00¦00¦00¦00¦00¦00¦00¦00¦00¦00¦00¦00¦00¦00¦
¦ @¦ @¦ @¦ @¦ @¦ @¦ @¦ @¦ @¦ @¦ @¦ @¦ @¦ @¦ @¦ @¦ @¦ @¦ @¦ @¦ @¦ @¦ @¦ @¦ @¦ @¦ @¦ @¦ @¦ @¦ @¦ @¦
```

BLOCK MAP 18/9

```
| 0| 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|27|28|29|30|31|
|12|11|83|14|01|70|72|65|66|65|72|65|6e|63|65|20|6d|67|72|a0|a0|14|0c|00|05|56|04|07|0c|00|16|00|
| r| q|-c| t| a| p| r| e| f| e| r| e| n| c| e|  | m| g| r|ss|ss| t| l| @| e| V| d| g| l| @| v| @|

|32|33|34|35|36|37|38|39|40|41|42|43|44|45|46|47|48|49|50|51|52|53|54|55|56|57|58|59|60|61|62|63|
|00|00|83|16|06|61|6c|61|72|6d|20|63|6c|6f|63|6b|a0|a0|a0|a0|a0|16|11|00|05|56|04|07|0c|00|0d|00|
| @| @|-c| v| f| a| l| a| r| m|  | c| l| o| c| k|ss|ss|ss|ss|ss| v| q| @| e| V| d| g| l| @| m| @|

|64|65|66|67|68|69|70|71|72|73|74|75|76|77|78|79|80|81|82|83|84|85|86|87|88|89|90|91|92|93|94|95|
|00|00|83|18|09|70|68|6f|74|6f|20|6d|61|6e|61|67|65|72|a0|a0|a0|18|01|00|05|56|04|07|0c|00|17|00|
| @| @|-c| x| i| p| h| o| t| o|  | m| a| n| a| g| e| r|ss|ss|ss| x| a| @| e| V| d| g| l| @| w| @|

            1
|96|97|98|99|00|01|02|03|04|05|06|07|08|09|10|11|12|13|14|15|16|17|18|19|20|21|22|23|24|25|26|27|
|00|00|83|19|0f|74|65|78|74|20|6d|61|6e|61|67|65|72|a0|a0|a0|a0|19|07|00|05|56|04|07|0c|00|14|00|
| @| @|-c| y| o| t| e| x| t|  | m| a| n| a| g| e| r|ss|ss|ss|ss| y| g| @| e| V| d| g| l| @| t| @|

|28|29|30|31|32|33|34|35|36|37|38|39|40|41|42|43|44|45|46|47|48|49|50|51|52|53|54|55|56|57|58|59|
|00|00|83|14|04|63|61|6c|63|75|6c|61|74|6f|72|a0|a0|a0|a0|a0|a0|14|0f|00|05|56|04|07|0c|00|10|00|
| @| @|-c| t| d| c| a| l| c| u| l| a| t| o| r|ss|ss|ss|ss|ss|ss| t| o| @| e| V| d| g| l| @| p| @|

|60|61|62|63|64|65|66|67|68|69|70|71|72|73|74|75|76|77|78|79|80|81|82|83|84|85|86|87|88|89|90|91|
|00|00|83|17|07|6e|6f|74|65|20|70|61|64|a0|a0|a0|a0|a0|a0|a0|a0|17|12|00|05|56|04|07|0c|00|11|00|
| @|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

                        2
|92|93|94|95|96|97|98|99|00|01|02|03|04|05|06|07|08|09|10|11|12|13|14|15|16|17|18|19|20|21|22|23|
|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

|24|25|26|27|28|29|30|31|32|33|34|35|36|37|38|39|40|41|42|43|44|45|46|47|48|49|50|51|52|53|54|55|
|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
```

Standard Commodore DOS file structure

With the exception of relative (REL) files, all standard DOS sequential (SEQ), program (PRG), and user (USR) files have very similar structures. The "BACKUP" program on the original GEOS disk is a good example.

A standard DOS filename contains only the file type byte in position 2, the pointer in positions 3 and 4, the name from positions 5 through 20, and the size in positions 30 and 31. If the filename is less than sixteen characters long, the remaining space for the filename in the directory entry is filled in, or padded, with shifted spaces [CHR$(160)]. The pointer indicates where on the disk the file begins. The Commodore DOS ignores any other characters that may be in positions 21 through 29.

"BACKUP" is a locked PRG file sixteen blocks long. The program actually begins on 19/0, as indicated by the pointer after the file type byte.

At 19/0, the first two bytes are again a pointer. This pointer indicates where the next block of the file is stored on disk, namely 19/14. All the bytes from position 2 through 255 are information used by the computer. Because this is a program file, positions 2 and 3 give the address in the computer that the program was saved from.

Using the pointer in positions 0 and 1, you can follow the file from block to block. You will finally end up at 23/5, the last block of the "BACKUP" program file. Here, byte 0 contains a null ($00). This indicates to the DOS that this is the last block of a file. The next byte gives the last position within the block that is valid. In this case, the $14 (20) indicates that the $80 in position 20 is the last valid byte in the file. Everything beyond that is garbage.

To GEOS, all standard DOS files, with the exception of REL files, are sequential. This is because these files all share the same structure, as described above, even if the standard DOS treats them as different types. The structure of the files will usually be invisible to the user unless you select the File/info command. When this command is selected, deskTop will indicate the way the file is built, sequentially as a standard DOS file, or as a GEOS VLIR file.

The exception throughout all of this is the standard DOS relative (REL) file. It is recommended that you do NOT try to construct a REL file on a GEOS disk, or worse, convert to GEOS format a disk that has a REL file on it. This area is little tested. If you use REL files, usually for a database program, it is a good idea to dedicate an entire disk to that file.

GEOS SEQUENTIAL FILES

GEOS sequential files are functionally identical to standard DOS files. Positions 3 and 4 of the directory entry point directly to the beginning of the file data, and the data blocks are linked sequentially until terminated with a null byte in position zero. GEOS sequential files can be identified by the byte in position 23. If position 23 contains a null ($00), the file structure is sequential (e.g. desk accessories, scraps, and input and printer drivers). If position 23 contains a $01, the file follows the VLIR structure, explained below.

GEOS INFORMATION BLOCK

All GEOS files also have an extra block, not a part of the actual file data, which contains information used by GEOS. Bytes 21 and 22 of the file directory entry point to the information block. All information blocks have the same structure, as follows:

Positions	Contain
0-1	$00, $ff. End of file indicator.
2-4	$03, $15, $bf. Size of sprite icon in characters wide, raster lines high, then color.
5-67	icon sprite data.
68	DOS file type (usually $83 or $c3).
69-70	GEOS file type bytes, order reversed from directory.
71-72	Load address.
73-74	End of file.
75-76	File entry point.
77-96	Class of file, e.g., Write Image V1.1.
97-116	Author.
117-136	Source program, e.g., geoWrite V1.1.
137-159	Use unknown, but usually constant within a GEOS file type.
160-255	File information box text, terminated by a $00. Maximum of 94 text characters, plus null.

geoWrite information block

```
| 0| 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|27|28|29|30|31|
|00|ff|03|15|bf|ff|ff|ff|80|00|01|80|01|f1|80|07|31|80|1c|61|80|30|c1|80|61|81|80|c3|01|81|a6|01|
| @|  | c| u|-?|  |  |  |-@| @| a|-@| a|  |-@| g| 1|-@|  | a|-@| 0|-A|-@| a|-a|-@|-C| a|-a|-&| a|
      |        |

|32|33|34|35|36|37|38|39|40|41|42|43|44|45|46|47|48|49|50|51|52|53|54|55|56|57|58|59|60|61|62|63|
|83|4c|01|82|b8|01|86|60|01|85|c1|e1|8b|02|11|8c|01|e1|88|02|f1|90|05|f9|90|f7|f9|af|83|f1|80|00|
|-c| L| a|-b|-8| a|-f|  | a|-e|-A|  |-k| b| q|-l| a|  |-h| b|  |-p| e|  |-p|  |  |-/|-c|  |-@| @|
                          sprite data (cont.)

|64|65|66|67|68|69|70|71|72|73|74|75|76|77|78|79|80|81|82|83|84|85|86|87|88|89|90|91|92|93|94|95|
|01|ff|ff|ff|83|06|01|00|04|ff|03|16|27|67|65|6f|57|72|69|74|65|20|20|20|20|56|31|2e|31|00|00|00|
| a|  |  |  |-c| f| a| @| d|  | c| v| '| g| e| o| W| r| i| t| e|  |  |  |  | V| 1| .| 1| @| @| @|
sprite------|  |     |     |     |     |             class  of file
             |
|96|97|98|99|00|01|02|03|04|05|06|07|08|09|10|11|12|13|14|15|16|17|18|19|20|21|22|23|24|25|26|27|
|00|54|6f|6e|79|20|52|65|71|75|69|73|74|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|
| @| T| o| n| y|  | R| e| q| u| i| s| t| @| @| @| @| @| @| @| @| @| @| @| @| @| @| @| @| @| @| @|
   |              Author                                        |

|28|29|30|31|32|33|34|35|36|37|38|39|40|41|42|43|44|45|46|47|48|49|50|51|52|53|54|55|56|57|58|59|
|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|00|9f|
| @| @| @| @| @| @| @| @| @| @| @| @| @| @| @| @| @| @| @| @| @| @| @| @| @| @| @| @| @| @| @|<-|


|60|61|62|63|64|65|66|67|68|69|70|71|72|73|74|75|76|77|78|79|80|81|82|83|84|85|86|87|88|89|90|91|
|67|65|6f|57|72|69|74|65|20|69|73|20|61|20|57|59|53|49|57|59|47|20|77|6f|72|64|20|70|72|6f|63|65|
| g| e| o| W| r| i| t| e|  | i| s|  | a|  | W| Y| S| I| W| Y| G|  | w| o| r| d|  | p| r| o| c| e|
|             File/info comments text.
                         2
|92|93|94|95|96|97|98|99|00|01|02|03|04|05|06|07|08|09|10|11|12|13|14|15|16|17|18|19|20|21|22|23|
|73|73|6f|72|20|77|69|74|68|20|70|72|6f|70|6f|72|74|69|6f|6e|61|6c|20|73|70|61|63|69|6e|67|20|61|
| s| s| o| r|  | w| i| t| h|  | p| r| o| p| o| r| t| i| o| n| a| l|  | s| p| a| c| i| n| g|  | a|


|24|25|26|27|28|29|30|31|32|33|34|35|36|37|38|39|40|41|42|43|44|45|46|47|48|49|50|51|52|53|54|55|
|6e|64|20|67|72|61|70|68|69|63|20|73|75|70|70|6f|72|74|2e|00|00|00|00|03|00|05|00|09|00|0f|00|13|
| n| d|  | g| r| a| p| h| i| c|  | s| u| p| p| o| r| t| .| @| @| @| @| c| @| e| @| i| @| o| @| s|
             File/info comments text, ending with a null.   |
```

GEOS VLIR file structure

Many GEOS files are not built sequentially. Rather, the files have one or more parts that are associated by means of a special index created by GEOS. Such files are in many ways similar to DOS REL files. GEOS calls this structure VLIR, an acronym for something like Very Large Indexed Relative files. A good example is a geoWrite document.

Bytes 3 and 4 of a geoWrite document directory entry point to a side sector. The side sector contains nothing but more pointers. Each two byte pointer points to a sequential series of blocks that contain the text data for one page. So bytes 2 and 3 point to the data for page one; bytes 4 and 5 point to the data for page two; etc.

Side sectors are all one block long. Byte 0 contains a null and byte 1 contains an $ff. The significance of the remaining bytes are defined by the particular application. The bytes of all side sectors point to data for an application, but the pointers in a geoWrite side sector point to individual pages and graphics; the pointers in a geoPaint side sector point to different parts of the graphic page, album side sectors point its own pages.

Some GEOS system files also use the VLIR file structure, most notably deskTop, GEOS KERNAL, geoWrite, and geoPaint. When GEOS system files are segmented in this fashion, some parts of the file are contiguous, some parts load into odd areas, and other parts overlay each other.

geoWrite file structure

Once you have looked at the directory entry and the information block, the next task is to get to the actual file data. Bytes 3 and 4 point to the track and block of the side sector, respectively.

A geoWrite side sector is initially filled with alternating $00, $ff. A side sector may be no longer than one block long, so bytes 0 and 1 of the side sector are always $00, $ff. Bytes 2 and 3 then point to the data for the first page, bytes 4 and 5 point to the data for the second page, etc. If the geoWrite document is one page long, only the first pointer will be used. GeoWrite allows for sixty-four document pages of text, requiring sixty-four pointers. Thus, the pointers for the geoWrite document text data can occupy bytes 2 through 129.

Graphics data used within a geoWrite document are stored separately from the text data. A flag within the geoWrite text data indicates that a Photo Scrap is being used. The pointer to the data for the first graphic used in a geoWrite document is at bytes 130 and 131 in the geoWrite document side sector; the pointer for the second graphic is at bytes 132 and 133, etc. Thus geoWrite can use up to 64 graphics in one document by using the side sector bytes from 130 through 255.

geoWrite data structure

GeoWrite stores text as standard ASCII code - an upper case "A" is stored as a 65 ($41) and a lower case "a" is a 97 ($61). The geoWrite document file can contain quite a bit of other information about formatting, graphics, fonts, and enhancements.

Each page of a geoWrite document begins with twenty bytes of header information starting at byte 2. This header information determines the position of the margins and tabs. Bytes 2 and 3 give the distance in pixels of the left margin from the left edge of the virtual document which starts at 1.2 inches on the ruler.

If the left margin is set at 1.2 inches on the ruler, the values in positions 2 and 3 will be $00, $00. If the left margin is set at 1.5 on the ruler, which is three-tenths of an inch from the left edge of the virtual document page, byte two of the first block of the document page file will contain a $18 (+24). This continues the assumption that you will be using an 80 dpi printer. (24 dots x 1 inch/80 dots = 24/80 = 3/10 inch.) The second byte is a high byte, and if it is equal to 1, then add 256 to the number of dots from the left margin.

The two bytes for the right margin are in positions 4 and 5. The values here default to $df (+223) and $01 (+1), respectively. Added up, it puts the right margin 479 dots, one dot less than 6 inches, to the right of the left edge of the print, or at the 7.2 inch mark on the ruler.

The next sixteen pairs of bytes give information for the tabs in the same format. If a tab is not used, it is set equal to the right margin. Finally, we get to the information about the text.

Byte 22 will always have a value of $17 (+23). This character is used to flag font information. Bytes 23 through 25 give the specific font, size, and enhancement information as described below. Actual text begins in position 26.

Text data for each page follows sequentially from one block to the next, as indicated by the pointers in positions 0 and 1. Carriage returns are standard ASCII CHR$(13)s, and word wrap is handled internally by geoWrite. If a page is vertically filled, the end of page will not be indicated within the disk data except by the pointer in the last block. In this case, the last block of text data for the page will have a $00 in position 0, and position 1 will indicate the last valid byte of the current block.

If the page is the last page of a document, the end of the file will be indicated in the last block of data for that page by a null.

If the page is not the last page, and is not vertically filled, then it ought to have a page break character in the last byte position for the page, as indicated by the pointer of the last block of text data. The page break character is the same as the ASCII form feed character, CHR$(12) ($0c).

If a Photo Scrap has been used in a geoWrite document, its position within the file will be flagged with a $10 (+16). The next three bytes will give the size of the Photo Scrap, followed by a key to the location of the graphics data on disk.

The first graphic within a file will have a $40 (+64) in the fourth position after the $10 flag. The second graphic will have a $41 (+65), etc. These bytes are used by geoWrite to find the pointer within the side sector to the graphics data on disk. Pointers to the graphics data are held in the side sector starting at position 130. From within the text data you can determine the position of the pointer by the formula:

position = (byte*2) + 2

For example, the position within the side sector of the pointer to the data for the third Photo Scrap used within a geoWrite document would be (66*2)+2) = 134. Thus bytes 134 and 135 point to the first block of the graphic data used in the third Photo Scrap graphic used in geoWrite.

Fonts and Enhancements

The actual graphics data that appears on the screen in geoWrite is not stored on disk with the geoWrite document file. Rather, the document file contains the ASCII values of the characters and flags that indicate the fonts and enhancements. Graphics data for imported Photo Scraps is loaded from disk.

Every change of font, or even change of enhancement, causes geoWrite to place within the document four bytes which completely describe the newly effective combination. Each set of four bytes begins with a $17 (+23) which flags to GEOS that the next three bytes describe the new font and style. The next two bytes determine the font, and the last byte determines the enhancements.

Of the two bytes that determine the font, the low order nybble (bits 0 through 3) of the first byte gives the size in points. If the size exceeds 15 points, the bits carry over into the high order nybble (bits 4 through 7). For example, all ten point fonts have bits 1 and 3 set of the first byte following the $17 flag. All eighteen point fonts have bits 4 and 1 set. Thus the size is directly indicated.

(An eight digit binary number with digits one and three set to 1 is equal to a decimal value of +10. If digits one and four are set, the value is equal to +18. A binary digit is referred to as a bit, and bits are numbered from the right, starting at zero. Position zero represents a value of 0 or 1, position one represents a value of 0 or 2, position three represents a value of 0 or 4, etc.)

The high order nybble of the first byte following the flag and the second byte following the flag determine the typeface. Because the size of the font can carry over from the low order nybble, the typeface must be determined by looking at the individual bits. For example, BSW has no bits of the high order nybble set, all California fonts have bit 7 set, and all Roma fonts have bits 7 and 6 set. The table on the next page gives a complete breakdown of the fonts and sizes.

FONT	SIZE	HEX	BINARY
BSW	9	09 00	0000 1001
Californ	10	8a 00	1000 1010
	12	8c 00	1000 1100
	14	8e 00	1000 1110
	18	92 00	1001 0010
Cory	12	4c 01	0100 1100 0000 0001
	24	58 01	0101 1000 0000 0001
Dwinelle	18	12 01	0001 0010 0000 0001
Roma	9	c9 00	1100 1001
	12	cc 00	1100 1100
	18	d2 00	1101 0010
	24	d8 00	1101 1000
Universi	6	46 00	0100 0110
	10	4a 00	0100 1010
	12	4c 00	0100 1100
	14	4e 00	0100 1110
	18	52 00	0101 0010
	24	58 00	0101 1000

The additional fonts available in Berkeley Softwork's FONTPAK1 will probably make more extensive use of the second byte.

The third byte following the $17 flag determines the enhancements active. A change of enhancements within the text will cause geoWrite to insert the full four bytes. Each enhancement corresponds to one bit of the third byte, as follows:

BIT	INDICATES
0	- -
1	- -
2	- -
3	outline
4	italic
5	- -
6	bold
7	underline

First Block of a geoWrite page containing Photo Scrap and Change of Font

```
| 0| 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|27|28|29|30|31|
|00|f4|00|00|df|01|28|00|df|01|df|01|df|01|df|01|df|01|df|01|df|01|17|09|00|00|54|68|69|73|20|66|
| @|  | @| @|  | a| (| @|  | a|  | a|  | a|  | a|  | a|  | a|  | a| w| i| @| @| T| h| i| s|  | f|

|32|33|34|35|36|37|38|39|40|41|42|43|44|45|46|47|48|49|50|51|52|53|54|55|56|57|58|59|60|61|62|63|
|69|6c|65|20|62|65|67|69|6e|73|20|77|69|74|68|20|73|74|61|6e|64|61|72|64|20|42|53|57|20|39|20|70|
| i| l| e|  | b| e| g| i| n| s|  | w| i| t| h|  | s| t| a| n| d| a| r| d|  | B| S| W|  | 9|  | p|

|64|65|66|67|68|69|70|71|72|73|74|75|76|77|78|79|80|81|82|83|84|85|86|87|88|89|90|91|92|93|94|95|
|6f|69|6e|74|20|61|6e|64|20|73|74|61|6e|64|61|72|64|20|6d|61|72|67|69|6e|73|20|61|6e|64|20|74|61|
| o| i| n| t|  | a| n| d|  | s| t| a| n| d| a| r| d|  | m| a| r| g| i| n| s|  | a| n| d|  | t| a|

             1
|96|97|98|99|00|01|02|03|04|05|06|07|08|09|10|11|12|13|14|15|16|17|18|19|20|21|22|23|24|25|26|27|
|62|73|2e|0d|17|4c|01|00|3c|2d|20|66|6f|75|72|20|62|79|74|65|73|20|69|6e|64|69|63|61|74|69|6e|67|
| b| s| .|^m|^w| L|^a| @| <| -|  | f| o| u| r|  | b| y| t| e| s|  | i| n| d| i| c| a| t| i| n| g|

|28|29|30|31|32|33|34|35|36|37|38|39|40|41|42|43|44|45|46|47|48|49|50|51|52|53|54|55|56|57|58|59|
|20|43|6f|72|79|20|31|32|20|70|6f|69|6e|74|17|cc|00|90|3c|20|2d|20|66|6f|75|72|20|62|79|74|65|73|
|  | C| o| r| y|  | 1| 2|  | p| o| i| n| t|^w|-L| @|-p| <|  | -|  | f| o| u| r|  | b| y| t| e| s|

|60|61|62|63|64|65|66|67|68|69|70|71|72|73|74|75|76|77|78|79|80|81|82|83|84|85|86|87|88|89|90|91|
|20|69|6e|64|69|63|61|74|69|6e|67|20|52|6f|6d|61|20|31|32|20|70|6f|69|6e|74|20|69|74|61|6c|69|63|
|  | i| n| f| i| c| a| t| i| n| g|  | R| o| m| a|  | 1| 2|  | p| o| i| n| t|  | i| t| a| l| i| c|

                         2
|92|93|94|95|96|97|98|99|00|01|02|03|04|05|06|07|08|09|10|11|12|13|14|15|16|17|18|19|20|21|22|23|
|20|75|6e|64|65|72|6c|69|6e|65|20|20|20|50|68|6f|74|6f|20|53|63|72|61|70|20|20|2d|2d|3e|10|16|68|
|  | u| n| d| e| r| l| i| n| e|  |  |  | P| h| o| t| o|  | S| c| r| a| p|  |  | -| -| >|^p|^v| h|

|24|25|26|27|28|29|30|31|32|33|34|35|36|37|38|39|40|41|42|43|44|45|46|47|48|49|50|51|52|53|54|55|
|00|40|70|61|67|65|20|20|62|72|65|61|6b|20|20|2d|20|2d|20|3e|0c|00|00|00|00|00|00|00|00|00|00|00|
| @|  | p| a| g| e|  |  | b| r| e| a| k|  |  | -|  | -|  | >|^l| @| @| @| @| @| @| @| @| @| @| @|
```

geoPaint file structure

GeoPaint document files have the same structure for the directory entry and information block as do geoWrite document files. The side sector and the data are different.

The geoPaint document side sector contains the standard $00, $ff in positions 0 and 1. If the document is empty, the bytes from positions 2 through 91 will be filled with alternating $00, $ff. All bytes from position 92 through 255 will have nulls.

The ninety bytes from positions 2 through 91 comprise forty-five pointers to actual geoPaint data. The data is stored sequentially on disk.

A geoPaint document consists of ninety lines of 8x8 pixel cards as seen on the monitor screen. These lines extend the width of the document page, not just the width of the drawing window. Each block of data indicated by the pointers in the side sector describe two of the ninety lines. If one of the two lines contains any data, geoWrite will save the data to disk and set the pointer in the appropriate position in the side sector.

If you have one pixel set in the extreme upper left of the document, and one pixel set in the extreme lower right of the document, then the geoWrite side sector will have a pointer set in position 2-3 and one in position 90-91. Text within a geoWrite document is saved as graphic data.

geoPaint data structure

GeoPaint compacts the graphics data that is used in a document. Unless the geoPaint document is exceedingly complex - unless virtually every graphic card is different from the one preceding it - the method of compacting the data makes geoPaint document files smaller than they would be otherwise. The method of compaction is at once simple and complex. It is fairly simple if you know how it works, but it is the very devil to figure out from scratch. But first a bit of background on Commodore high resolution (hi-res) graphics in general.

The video output of the Commodore 64 is controlled by the VIC II chip. (The VIC 20 uses the 6567 Video Interface Chip [VIC], the C64 uses the 6560 or 6561 VIC II, and the C128 uses the 8564 VIC [3?] chip. Hence the name of the VIC series. All three work similarly.) The VIC chip works the C64 screen in terms of 8x8 pixel cards forty across by twenty-five down. This allows one thousand character blocks, or 320x200 pixels for graphics. But even when working with graphics, the VIC chip works in terms of these 8x8 pixel cards.

The cards are broken down into eight horizontal rows of eight pixels each. Each of these rows of eight pixels is controlled by one byte, and each pixel within the row is controlled by one bit of that byte. The VIC chip requires eight bytes to define each 8x8 pixel graphics card, and the screen has one thousand such cards. Thus, a complete hi-res graphics screen requires 8000 bytes of information to control each of the 64,000 pixels on the screen. A geoPaint document has about 512,000 possible pixels on the document page. If the graphics data were not compacted, a geoPaint document would occupy 64,000 bytes on disk!

When working with hi-res graphics, both bits and bytes are numbered from zero. The chart on the following page shows how a single hi-res graphics card might look.

graphics card

	column 1									
	7	6	5	4	3	2	1	0	=	Byte number
	0	0	1	1	1	1	0	0	$18	0
	0	1	0	0	0	0	1	0	$42	1
	0	1	0	0	0	0	1	0	$42	2
row 1	1	0	0	0	0	0	0	1	$81	3
	1	0	1	0	0	1	0	1	$a5	4
	0	1	0	1	1	0	1	0	$5a	6
	0	1	0	0	0	0	1	0	$42	6
	0	0	1	1	1	1	0	0	$3c	7
	0	0	1	1	1	1	0	0	$18	320
	0	1	0	0	0	0	1	0	$42	321
	0	1	0	0	0	0	1	0	$42	322
row 2	1	0	0	0	0	0	0	1	$81	323
	1	0	1	0	0	1	0	1	$a5	324
	0	1	0	1	1	0	1	0	$5a	325
	0	1	0	0	0	0	1	0	$42	326
	0	0	1	1	1	1	0	0	$3c	327

Because all video of the C64 goes through the VIC chip, geoPaint graphics must be able to be translated into similar graphics cards. The two cards above would look something like one smiley face on top of another.

GeoPaint document data is encrypted to save space. It would be exceedingly difficult to give an example of a geoPaint document page, then list the data to show how the data is encrypted. We must instead give an explanation of the geoPaint graphics data encryption scheme, plus a few small examples of individual cards and two line segments. The encryption is not done to make life difficult for the programmer, but to make life easier for the user.

GeoPaint works with lines that run the full width of the geoPaint document page, or 640 dots, two lines at a time. That is, each pointer in a geoPaint side sector points to the data for two rows of cards on the screen. The data starts in position 2 of the first block indicated by the side sector, beginning with the graphic data for the full two lines, followed by the color data for the full two lines. The end of the two line section is indicated by a binary zero. GeoPaint does not use any delimiters within the data file on disk, either between one graphics definition and the next, or between graphics data and color data. Rather, it appears to separate the two by counting within the computer.

Although two full lines are completely defined by 160 cards of graphics data, geoPaint defines 161 cards. The one hundred and sixty-first card is almost always defined as a blank card ($88 $00), but because it can so easily be preceded by an identical card is not sufficient to serve as a delimiter.

GeoPaint has two methods of encrypting and one way to use literal graphics data for storage on disk. The first method of encryption compacts repeating byte patterns. This encryption uses two bytes, the first as a counter and the second as the byte pattern to be repeated.

The counter for this type of encryption indicates that the first type of encryption is being used by having the high bit set. Thus, all values for this method of encryption will be between $80 (+128) and $ff (+255). This high bit must be masked off to give the actual number of times (1-127) that the next byte will be repeated.

For example, the two bytes $88 $ff represent one card with all pixels on. The second byte, $ff, is the pattern that is repeated the number of times indicated by the first byte. The first byte shows that the bit pattern given by the second byte repeats 8 times by the following formula.

$88 = %1000 1000
    mask off the high (leftmost) bit to yield
$08 = %0000 1000 = 8

Thus the bit pattern given by the second byte repeats eight times to define one full graphic card.

The example below shows how 72 blank cards might look within a geoPaint document file.

```
$ff $00       $ff $00        $ff $00        $ff $00        $c4 $00
 |   |                                                      |
 |   |-> %0000 0000 all pixels in one row off               |
 |-> %1111 1111 and take off the high                       |
               bit to yield                                 |
               %0111 1111 = $80 = +127                      |
                                                            |
                   $c4 = %11000100 and take off the high <--|
                   bit to yield %0100 0100 = $44 = + 68
```

Here we have ten bytes broken down into five pairs, and the first byte of each pair has the high bit set to indicate that the first encryption method is being used. Masking off the high bit for each of the first four pairs yields a value of +127 ($7f) for each of them. Masking off the high bit for the first byte ($c4) of the last pair yields a +68 ($44). All the bytes that show the bit pattern in this example are $00, indicating that the line of the graphics card is blank.

Adding the value of the masked counters, we get the sum

127 + 127 + 127 + 127 + 68 = 576

total bytes defined. And, as each card uses eight bytes to define it, 576 bytes breaks down to 72 cards (576/8 = 72) defined.

The second method of graphics data encryption is indicated by bit six, but not bit seven, being set in the counter byte. Thus it must be in the range of $40 (+64) through $7f (+127). In practice, the values $40 and $41 are never used. The modified counter indicates the number of times the pattern defined by the next eight bytes is repeated. For example,

$6c = %0110 1100 -> %0010 1100 = $2c = +44

indicates that the following pattern will be used +44 times.

The pattern that follows a counter in the range of $42 to $7f is eight bytes long. The eight bytes are sufficient to define one graphics card. This method of graphics data encryption is particularly useful for compacting geoPaint fill patterns or empty space. The following sequence

61 c3 66 3c 18 18 3c 66 c3

would indicate the eight byte pattern that looks something like an "X" would be repeated thirty-three times (the number of cards in a geoWrite working window). [$61 = 0110 0001 -> 0010 0001 = $21 = +33]

```
1 1 0 0 0 0 1 1   =  $c3
0 1 1 0 0 1 1 0   =  $66
0 0 1 1 1 1 0 0   =  $3c
0 0 0 1 1 0 0 0   =  $18
0 0 0 1 1 0 0 0   =  $18
0 0 1 1 1 1 0 0   =  $3c
0 1 1 0 0 1 1 0   =  $66
1 1 0 0 0 0 1 1   =  $c3
```

Finally, geoPaint will use literal graphics data if there is no eight bit (one byte) pattern that repeats four or more times. In this case, the literal graphics data will be flagged with a value in the range of $01 to $3f (+63). In this case, no bits are stripped, and the counter directly indicates the number of following bytes to be used literally. Each byte defines one line of a card.

COLOR DATA

The color data for the geoPaint hi-res graphics immediately follows definition of the one hundred and sixty-first card of the graphics data. The foreground (brush) and background (canvas) colors are defined in one byte broken down into two nybbles. The first nybble defines the foreground color, the second defines the background color. The default value $bf indicates a foreground color of light grey ($b = +11 -> light grey) and background color of dark grey ($f = +15 -> dark grey). The value $67 would indicate a foreground color of blue and a background color of yellow.

Color data is also encrypted in the same fashion as the graphics data. Thus, seventy-seven cards with a green foreground on a black background would be indicated by

```
cd 50  ($cd = %1100 1101 -> %0100 1101 = +77)
   ||-> black
   |--> green
```

The following example shows how geoPaint would store a very simple graphic. In this case, the first seven 8 bit rows of the first card are set to the foreground color, the last row cleared to the background color, the top row of the second card cleared to background, and the second row (byte) of the second card set to foreground. The remainder of the section defined by this block is cleared to the background color. Finally, the five bytes at the end give the color for all the cards and then delimit the end of the section with a null.

Two lines of graphic data, with color

```
87 ff 03 00 00 ff ff 00 ff 00 ff 00 ff 00 ff
  ¦     ¦         ¦-------  1278 $00s ------
  ¦     ¦-> two $00s and one $ff
  ¦-> seven $ff

 00 ff 00 ff 00 ff 00 ff 00 88 00 ff bf a1 bf 00
 ---- $00s (cont.) ------------>¦ ¦- color -¦ end

GRAPHICS
    $87 = %1000 0111 -> %0000 0111 = 7
    7 + 2 + 1 + 1278 = 1288 bytes defined
    1288/8 = 161 graphic cards defined, 160 cards used.
COLOR
    $ff = %1111 1111 -> %0111 1111 = 127
    $a1 = %1010 0001 -> %0010 0001 = _33
                                     160 cards with color $bf
```

GEOS DISK MAP

This book is not large enough to give all the interesting information that GEOS may yield. You can learn much from the data on the GEOS disk, if you know where to look. On the following pages is given a map of the GEOS disk that names the contents and shows the pointer for each block of the GEOS V1.2 disk as shipped by Berkeley Softworks during the summer and fall of 1986.

Tracks and sectors are given first in decimal, then in hexadecimal. The name of the file of which the block of the part is given next. If the file follows the VLIR file structure, the number after the file name indicates which pair of bytes in the side sector point to the start of this segment of the file. Then is given the block's position within the sequence. Finally, again in the case of VLIR files, the position relative to the first block of the first segment is given.

Blocks which are side sectors are further detailed with a list of each of the side sector pointers.

For example, the following line shows that 2/0 (track 2, block 0) is part of the GEOS Kernal file. Specifically, it is the seventh block of the sequence that is indicated by the pointer in positions 4 and 5 of the GEOS Kernal side sector. It is the twenty-fourth block of the file as a whole, including the first sequence, and the next block in the current sequence is at 2/8.

2	0	Geos Kernal #2, block 7 (24)	2, 8

Track	Sector	Contents	Link
1	0	Geos Kernal #1, block 7	1,15
1	1	Geos Kernal #1, block 5	1,13
1	2	Geos Kernal #1, block 9	1,16
1	3	Geos Kernal #1, block 11	1, 5
1	4	Geos Boot, block 2	1,12
1	5	Geos Kernal #1, block 12	2,14
1	6	Geos Kernal #1, block 3	1,14
1	7	Geos Boot, block 5 (last block of the file)	----
1	8	Geos	----
1	9	Geos Boot INFO BLOCK	----
1	10 $0a	Geos Kernal INFO BLOCK	----
1	11 $0b	Geos Kernal #1, block 1	1,19
1	12 $0c	Geos Boot, block 3	1,20
1	13 $0d	Geos Kernal #1, block 6	1, 0
1	14 $0e	Geos Kernal #1, block 4	1, 1
1	15 $0f	Geos Kernal #1, block 8	1, 2
1	16 $10	Geos Kernal #1, block 10	1, 3
1	17 $11	Geos Boot, block 1 (no side sector)	1, 4
1	18 $12	Geos Kernal SIDE SECTOR 1,11 (17 blocks) 2,15 (66 blocks)	----
1	19 $13	Geos Kernal #1, block 2	1, 6
1	20 $14	Geos Boot, block 4	1, 7
2	0	Geos Kernal #2, block 7 (24)	2, 8
2	1	Geos Kernal #1, block 14	2, 9
2	2	Geos Kernal #2, block 2 (19)	2,10
2	3	Geos Kernal #2, block 10 (27)	2,11
2	4	Geos Kernal #1, block 17 (last block of part #1)	----
2	5	Geos Kernal #2, block 5 (22)	2,13
2	6	Geos Kernal #2, block 13 (30)	2,20
2	7	Geos Kernal #2, block 15 (32)	2,12
2	8	Geos Kernal #2, block 8 (25)	2,16
2	9	Geos Kernal #1, block 15	2,17
2	10 $0a	Geos Kernal #2, block 3 (20)	2,18
2	11 $0b	Geos Kernal #2, block 11 (28)	2,19
2	12 $0c	Geos Kernal #2, block 16 (33)	3,14
2	13 $0d	Geos Kernal #2, block 6 (23)	2, 0
2	14 $0e	Geos Kernal #1, block 13	2, 1
2	15 $0f	Geos Kernal #2, block 1 (18)	2, 2
2	16 $10	Geos Kernal #2, block 9 (26)	2, 3
2	17 $11	Geos Kernal #1, block 16	2, 4
2	18 $12	Geos Kernal #2, block 4 (21)	2, 5
2	19 $13	Geos Kernal #2, block 12 (29)	2, 6
2	20 $14	Geos Kernal #2, block 14 (31)	2, 7

Track	Sector	Contents	Link
3	0	Geos Kernal #2, block 31 (48)	3, 8
3	1	Geos Kernal #2, block 18 (35)	3, 9
3	2	Geos Kernal #2, block 26 (43)	3,10
3	3	Geos Kernal #2, block 34 (51)	3,11
3	4	Geos Kernal #2, block 21 (38)	3,12
3	5	Geos Kernal #2, block 29 (46)	3,13
3	6	Geos Kernal #2, block 37 (54)	4,14
3	7	Geos Kernal #2, block 24 (41)	3,15
3	8	Geos Kernal #2, block 32 (49)	3,16
3	9	Geos Kernal #2, block 19 (36)	3,17
3	10 $0a	Geos Kernal #2, block 27 (44)	3,18
3	11 $0b	Geos Kernal #2, block 35 (52)	3,19
3	12 $0c	Geos Kernal #2, block 22 (39)	3,20
3	13 $0d	Geos Kernal #2, block 30 (47)	3, 0
3	14 $0e	Geos Kernal #2, block 17 (34)	3, 1
3	15 $0f	Geos Kernal #2, block 25 (42)	3, 2
3	16 $10	Geos Kernal #2, block 33 (50)	3, 3
3	17 $11	Geos Kernal #2, block 20 (37)	3, 4
3	18 $12	Geos Kernal #2, block 28 (45)	3, 5
3	19 $13	Geos Kernal #2, block 36 (53)	3, 6
3	20 $14	Geos Kernal #2, block 23 (40)	3, 7
4	0	Geos Kernal #2, block 52 (69)	4, 8
4	1	Geos Kernal #2, block 39 (56)	4, 9
4	2	Geos Kernal #2, block 47 (64)	4,10
4	3	Geos Kernal #2, block 55 (72)	4,11
4	4	Geos Kernal #2, block 42 (59)	4,12
4	5	Geos Kernal #2, block 50 (67)	4,13
4	6	Geos Kernal #2, block 58 (75)	5,14
4	7	Geos Kernal #2, block 45 (62)	4,15
4	8	Geos Kernal #2, block 53 (70)	4,16
4	9	Geos Kernal #2, block 40 (57)	4,17
4	10 $0a	Geos Kernal #2, block 48 (65)	4,18
4	11 $0b	Geos Kernal #2, block 56 (73)	4,19
4	12 $0c	Geos Kernal #2, block 43 (60)	4,20
4	13 $0d	Geos Kernal #2, block 51 (68)	4, 0
4	14 $0e	Geos Kernal #2, block 38 (55)	4, 1
4	15 $0f	Geos Kernal #2, block 46 (63)	4, 2
4	16 $10	Geos Kernal #2, block 54 (71)	4, 3
4	17 $11	Geos Kernal #2, block 41 (58)	4, 4
4	18 $12	Geos Kernal #2, block 49 (66)	4, 5
4	19 $13	Geos Kernal #2, block 57 (74)	4, 6
4	20 $14	Geos Kernal #2, block 44 (61)	4, 7

Track	Sector	Contents	Link
5	0	Desk Top INFO BLOCK	----
5	1	Geos Kernal #2, block 60 (77)	5, 9
5	2	Desk Top #1, block 7	5,10
5	3	Desk Top #1, block 2	5,11
5	4	Geos Kernal #2, block 63 (80)	5,12
5	5	Desk Top #1, block 10	5,13
5	6	Desk Top #1, block 5	5,15
5	7	Geos Kernal #2, block 66 (83-last file block)	----
5	8	Desk Top SIDE SECTOR:	----
		5,16 (62 blocks)	
		8, 2 (8 blocks)	
5	9	Geos Kernal #2, block 61 (78)	5,17
5	10 $0a	Desk Top #1, block 8	5,18
5	11 $0b	Desk Top #1, block 3	5,19
5	12 $0c	Geos Kernal #2, block 64 (81)	5,20
5	13 $0d	Desk Top #1, block 11	6,14
5	14 $0e	Geos Kernal #2, block 59 (76)	5, 1
5	15 $0f	Desk Top #1, block 6	5, 2
5	16 $10	Desk Top #1, block 1	5, 3
5	17 $11	Geos Kernal #2, block 62 (79)	5, 4
5	18 $12	Desk Top #1, block 9	5, 5
5	19 $13	Desk Top #1, block 4	5, 6
5	20 $14	Geos Kernal #2, block 65 (82)	5, 7
6	0	Desk Top #1, block 26	6, 8
6	1	Desk Top #1, block 13	6, 9
6	2	Desk Top #1, block 21	6,10
6	3	Desk Top #1, block 29	6,11
6	4	Desk Top #1, block 16	6,12
6	5	Desk Top #1, block 24	6,13
6	6	Desk Top #1, block 32	7,14
6	7	Desk Top #1, block 19	6,15
6	8	Desk Top #1, block 27	6,16
6	9	Desk Top #1, block 14	6,17
6	10 $0a	Desk Top #1, block 22	6,18
6	11 $0b	Desk Top #1, block 30	6,19
6	12 $0c	Desk Top #1, block 17	6,20
6	13 $0d	Desk Top #1, block 25	6, 0
6	14 $0e	Desk Top #1, block 12	6, 1
6	15 $0f	Desk Top #1, block 20	6, 2
6	16 $10	Desk Top #1, block 28	6, 3
6	17 $11	Desk Top #1, block 15	6, 4
6	18 $12	Desk Top #1, block 23	6, 5
6	19 $13	Desk Top #1, block 31	6, 6
6	20 $14	Desk Top #1, block 18	6, 7

Track	Sector	Contents		Link
7	0	Desk Top #1, block 47		7, 8
7	1	Desk Top #1, block 34		7, 9
7	2	Desk Top #1, block 42		7,10
7	3	Desk Top #1, block 50		7,11
7	4	Desk Top #1, block 37		7,12
7	5	Desk Top #1, block 45		7,13
7	6	Desk Top #1, block 53		8,14
7	7	Desk Top #1, block 40		7,15
7	8	Desk Top #1, block 48		7,16
7	9	Desk Top #1, block 35		7,17
7	10 $0a	Desk Top #1, block 43		7,18
7	11 $0b	Desk Top #1, block 51		7,19
7	12 $0c	Desk Top #1, block 38		7,20
7	13 $0d	Desk Top #1, block 46		7, 0
7	14 $0e	Desk Top #1, block 33		7, 1
7	15 $0f	Desk Top #1, block 41		7, 2
7	16 $10	Desk Top #1, block 49		7, 3
7	17 $11	Desk Top #1, block 36		7, 4
7	18 $12	Desk Top #1, block 44		7, 5
7	19 $13	Desk Top #1, block 52		7, 6
7	20 $14	Desk Top #1, block 39		7, 7
8	0	Desk Top #2, block 6	(68)	8, 8
8	1	Desk Top #1, block 55		8, 9
8	2	Desk Top #2, block 1	(63)	8,10
8	3	Geopaint #1, block 1		13, 2
8	4	Desk Top #1, block 58		8,12
8	5	Desk Top #2, block 4		8,13
8	6	Geowrite INFO BLOCK		----
8	7	Desk Top #1, block 61		8,15
8	8	Desk Top #2, block 7	(69)	8,16
8	9	Desk Top #1, block 56		8,17
8	10 $0a	Desk Top #2, block 2	(64)	8,18
8	11 $0b	Geopaint INFO BLOCK		----
8	12 $0c	Desk Top #1, block 59		8,20
8	13 $0d	Desk Top #2, block 5	(67)	8, 0
8	14 $0e	Desk Top #1, block 54		8, 1
8	15 $0f	Desk Top #1, block 62	(last block of part one)	----
8	16 $10	Desk Top #2, block 8	(70-last block of file)	----
8	17 $11	Desk Top #1, block 57		8, 4
8	18 $12	Desk Top #2, block 3	(65)	8, 5
8	19 $13	Geopaint SIDE SECTOR:		----
			8, 3 (64 blocks)	
			16,17 (7 blocks)	
			16,10 (10 blocks)	
			16, 6 (12 blocks)	
			17,18 (11 blocks)	
			19, 5 (6 blocks)	
			19,15 (7 blocks)	
8	20 $14	Desk Top #1, block 60		8, 7

Track	Sector	Contents	Link
9	0	Geowrite #1, block 14	9, 8
9	1	Geowrite #1, block 1	9, 9
9	2	Geowrite #1, block 9	9,10
9	3	Geowrite #1, block 17	9,11
9	4	Geowrite #1, block 4	9,12
9	5	Geowrite #1, block 12	9,13
9	6	Geowrite #1, block 20	10,14
9	7	Geowrite #1, block 7	9,15
9	8	Geowrite #1, block 15	9,16
9	9	Geowrite #1, block 2	9,17
9	10 $0a	Geowrite #1, block 10	9,18
9	11 $0b	Geowrite #1, block 18	9,19
9	12 $0c	Geowrite #1, block 5	9,20
9	13 $0d	Geowrite #1, block 13	9, 0
9	14 $0e	Geowrite SIDE SECTOR:	----
		9, 1 (36 blocks)	
		10,16 (14 blocks)	
		11, 2 (10 blocks)	
		11,19 (7 blocks)	
		12,12 (7 blocks)	
		12, 5 (7 blocks)	
		12,19 (5 blocks)	
9	15 $0f	Geowrite #1, block 8	9, 2
9	16 $10	Geowrite #1, block 16	9, 3
9	17 $11	Geowrite #1, block 3	9, 4
9	18 $12	Geowrite #1, block 11	9, 5
9	19 $13	Geowrite #1, block 19	9, 6
9	20 $14	Geowrite #1, block 6	9, 7
10 0a	0	Geowrite #1, block 35	10, 8
10 0a	1	Geowrite #1, block 22	10, 9
10 0a	2	Geowrite #1, block 30	10,10
10 0a	3	Geowrite #2, block 2 (38)	10,11
10 0a	4	Geowrite #1, block 25	10,12
10 0a	5	Geowrite #1, block 33	10,13
10 0a	6	Geowrite #2, block 5 (41)	11,14
10 0a	7	Geowrite #1, block 28	10,15
10 0a	8	Geowrite #1, block 36 (last block of part one)	----
10 0a	9	Geowrite #1, block 23	10,17
10 0a	10 $0a	Geowrite #1, block 31	10,18
10 0a	11 $0b	Geowrite #2, block 3 (39)	10,19
10 0a	12 $0c	Geowrite #1, block 26	10,20
10 0a	13 $0d	Geowrite #1, block 34	10, 0
10 0a	14 $0e	Geowrite #1, block 21	10, 1
10 0a	15 $0f	Geowrite #1, block 29	10, 2
10 0a	16 $10	Geowrite #2, block 1 (37)	10, 3
10 0a	17 $11	Geowrite #1, block 24	10, 4
10 0a	18 $12	Geowrite #1, block 32	10, 5
10 0a	19 $13	Geowrite #2, block 4 (40)	10, 6
10 0a	20 $14	Geowrite #1, block 27	10, 7

Track	Sector	Contents	Link
11 0b	0	Geowrite #3, block 6 (56)	11, 8
11 0b	1	Geowrite #2, block 7 (43)	11, 9
11 0b	2	Geowrite #3, block 1 (51)	11,10
11 0b	3	Geowrite #3, block 9 (59)	11,11
11 0b	4	Geowrite #2, block 10 (46)	11,12
11 0b	5	Geowrite #3, block 4 (54)	11,13
11 0b	6	Geowrite #4, block 2 (62)	12,14
11 0b	7	Geowrite #2, block 13 (49)	11,15
11 0b	8	Geowrite #3, block 7 (57)	11,16
11 0b	9	Geowrite #2, block 8 (44)	11,17
11 0b	10 $0a	Geowrite #3, block 2 (52)	11,18
11 0b	11 $0b	Geowrite #3, block 10 (60-last block of part 3)	-----
11 0b	12 $0c	Geowrite #2, block 11 (47)	11,20
11 0b	13 $0d	Geowrite #3, block 5 (55)	11, 0
11 0b	14 $0e	Geowrite #2, block 6 (42)	11, 1
11 0b	15 $0f	Geowrite #2, block 14 (50-last block of part 2)	-----
11 0b	16 $10	Geowrite #3, block 8 (58)	11, 3
11 0b	17 $11	Geowrite #2, block 9 (45)	11, 4
11 0b	18 $12	Geowrite #3, block 3 (53)	11, 5
11 0b	19 $13	Geowrite #4, block 1 (61)	11, 6
11 0b	20 $14	Geowrite #2, block 12 (48)	11, 7
12 0c	0	Geowrite #6, block 3 (77)	12, 8
12 0c	1	Geowrite #4, block 4 (64)	12, 9
12 0c	2	Geowrite #5, block 5 (72)	12,10
12 0c	3	Geowrite #6, block 6 (80)	12,11
12 0c	4	Geowrite #4, block 7 (67-last block of part 4)	-----
12 0c	5	Geowrite #6, block 1 (75)	12,13
12 0c	6	Geowrite #7, block 2 (83)	13,14
12 0c	7	Geowrite #5, block 3 (70)	12,15
12 0c	8	Geowrite #6, block 4 (78)	12,16
12 0c	9	Geowrite #4, block 5 (65)	12,17
12 0c	10 $0a	Geowrite #5, block 6 (73)	12,18
12 0c	11 $0b	Geowrite #6, block 7 (81-last block of part 6)	-----
12 0c	12 $0c	Geowrite #5, block 1 (68)	12,20
12 0c	13 $0d	Geowrite #6, block 2 (76)	12, 0
12 0c	14 $0e	Geowrite #4, block 3 (63)	12, 1
12 0c	15 $0f	Geowrite #5, block 4 (71)	12, 2
12 0c	16 $10	Geowrite #6, block 5 (79)	12, 3
12 0c	17 $11	Geowrite #4, block 6 (66)	12, 4
12 0c	18 $12	Geowrite #5, block 7 (74-last block of part 5)	-----
12 0c	19 $13	Geowrite #7, block 1 (82)	12, 6
12 0c	20 $14	Geowrite #5, block 2 (69)	12, 7

Track	Sector	Contents	Link
13 0d	0	Geopaint #1, block 7	13, 8
13 0d	1	Geowrite #7, block 4 (85)	13, 9
13 0d	2	Geopaint #1, block 2	13,10
13 0d	3	Geopaint #1, block 10	13,11
13 0d	4	Geopaint #1, block 15	13,12
13 0d	5	Geopaint #1, block 5	13,13
13 0d	6	Geopaint #1, block 13	13,15
13 0d	7	Geopaint #1, block 18	13,17
13 0d	8	Geopaint #1, block 8	13,16
13 0d	9	Geowrite #7, block 5 (86-last block of file)	-----
13 0d	10 $0a	Geopaint #1, block 3	13,18
13 0d	11 $0b	Geopaint #1, block 11	13,19
13 0d	12 $0c	Geopaint #1, block 16	13,20
13 0d	13 $0d	Geopaint #1, block 6	13, 0
13 0d	14 $0e	Geowrite #7, block 3 (84)	13, 1
13 0d	15 $0f	Geopaint #1, block 14	13, 4
13 0d	16 $10	Geopaint #1, block 9	13, 3
13 0d	17 $11	Geopaint #1, block 19	14,14
13 0d	18 $12	Geopaint #1, block 4	13, 5
13 0d	19 $13	Geopaint #1, block 12	13, 6
13 0d	20 $14	Geopaint #1, block 17	13, 7
14 0e	0	Geopaint #1, block 34	14, 8
14 0e	1	Geopaint #1, block 21	14, 9
14 0e	2	Geopaint #1, block 29	14,10
14 0e	3	Geopaint #1, block 37	14,11
14 0e	4	Geopaint #1, block 24	14,12
14 0e	5	Geopaint #1, block 32	14,13
14 0e	6	Geopaint #1, block 40	15,14
14 0e	7	Geopaint #1, block 27	14,15
14 0e	8	Geopaint #1, block 35	14,16
14 0e	9	Geopaint #1, block 22	14,17
14 0e	10 $0a	Geopaint #1, block 30	14,18
14 0e	11 $0b	Geopaint #1, block 38	14,19
14 0e	12 $0c	Geopaint #1, block 25	14,20
14 0e	13 $0d	Geopaint #1, block 33	14, 0
14 0e	14 $0e	Geopaint #1, block 20	14, 1
14 0e	15 $0f	Geopaint #1, block 28	14, 2
14 0e	16 $10	Geopaint #1, block 36	14, 3
14 0e	17 $11	Geopaint #1, block 23	14, 4
14 0e	18 $12	Geopaint #1, block 31	14, 5
14 0e	19 $13	Geopaint #1, block 39	14, 6
14 0e	20 $14	Geopaint #1, block 26	14, 7

Track	Sector	Contents	Link
15 0f	0	Geopaint #1, block 55	15, 8
15 0f	1	Geopaint #1, block 42	15, 9
15 0f	2	Geopaint #1, block 50	15,10
15 0f	3	Geopaint #1, block 58	15,11
15 0f	4	Geopaint #1, block 45	15,12
15 0f	5	Geopaint #1, block 53	15,13
15 0f	6	Geopaint #1, block 61	16,14
15 0f	7	Geopaint #1, block 48	15,15
15 0f	8	Geopaint #1, block 56	15,16
15 0f	9	Geopaint #1, block 43	15,17
15 0f	10 $0a	Geopaint #1, block 51	15,18
15 0f	11 $0b	Geopaint #1, block 59	15,19
15 0f	12 $0c	Geopaint #1, block 46	15,20
15 0f	13 $0d	Geopaint #1, block 54	15, 0
15 0f	14 $0e	Geopaint #1, block 41	15, 1
15 0f	15 $0f	Geopaint #1, block 49	15, 2
15 0f	16 $10	Geopaint #1, block 57	15, 3
15 0f	17 $11	Geopaint #1, block 44	15, 4
15 0f	18 $12	Geopaint #1, block 52	15, 5
15 0f	19 $13	Geopaint #1, block 60	15, 6
15 0f	20 $14	Geopaint #1, block 47	15, 7
16 10	0	Geopaint #3, block 5 (76)	16, 8
16 10	1	Geopaint #1, block 63	16, 9
16 10	2	Geopaint #2, block 7 (71-last block of part 2)	-----
16 10	3	Geopaint #3, block 8 (79)	16,11
16 10	4	Geopaint #2, block 2 (66)	16,12
16 10	5	Geopaint #3, block 3 (74)	16,13
16 10	6	Geopaint #4, block 1 (82)	17,14
16 10	7	Geopaint #2, block 5 (69)	16,15
16 10	8	Geopaint #3, block 6 (77)	16,16
16 10	9	Geopaint #1, block 64 (last block of part 1)	-----
16 10	10 $0a	Geopaint #3, block 1 (72)	16,18
16 10	11 $0b	Geopaint #3, block 9 (80)	16,19
16 10	12 $0c	Geopaint #2, block 3 (67)	16,20
16 10	13 $0d	Geopaint #3, block 4 (75)	16, 0
16 10	14 $0e	Geopaint #1, block 62	16, 1
16 10	15 $0f	Geopaint #2, block 6 (70)	16, 2
16 10	16 $10	Geopaint #3, block 7 (78)	16, 3
16 10	17 $11	Geopaint #2, block 1 (65)	16, 4
16 10	18 $12	Geopaint #3, block 2 (73)	16, 5
16 10	19 $13	Geopaint #3, block 10 (81-last block of part 3)	-----
16 10	20 $14	Geopaint #2, block 4 (68)	16, 7

Track	Sector	Contents	Link
17 11	0	Geopaint #5, block 4 (97)	17, 8
17 11	1	Geopaint #4, block 3 (84)	17, 9
17 11	2	Geopaint #4, block 11 (92)	17,10
17 11	3	Geopaint #5, block 7 (100)	17,11
17 11	4	Geopaint #4, block 6 (87)	17,12
17 11	5	Geopaint #5, block 2 (95)	17,13
17 11	6	Geopaint #5, block 10 (103)	19,16
17 11	7	Geopaint #4, block 9 (90)	17,15
17 11	8	Geopaint #5, block 5 (98)	17,16
17 11	9	Geopaint #4, block 4 (85)	17,17
17 11	10 $0a	Geopaint #4, block 12 (93-last block of part 4)	-----
17 11	11 $0b	Geopaint #5, block 8 (101)	17,19
17 11	12 $0c	Geopaint #4, block 7 (88)	17,20
17 11	13 $0d	Geopaint #5, block 3 (96)	17, 0
17 11	14 $0e	Geopaint #4, block 2 (83)	17, 1
17 11	15 $0f	Geopaint #4, block 10 (91)	17, 2
17 11	16 $10	Geopaint #5, block 6 (99)	17, 3
17 11	17 $11	Geopaint #4, block 5 (86)	17, 4
17 11	18 $12	Geopaint #5, block 1 (94)	17, 5
17 11	19 $13	Geopaint #5, block 9 (102)	17, 6
17 11	20 $14	Geopaint #4, block 8 (89)	17, 7
18 12	0	Block Availability Map (BAM)	18, 1
18 12	1	Directory block one	18, 9
18 12	2	----- unused -----	-----
18 12	3	----- unused -----	-----
18 12	4	----- unused -----	-----
18 12	5	----- unused -----	-----
18 12	6	Directory block four	-----
18 12	7	----- unused -----	-----
18 12	8	----- unused -----	-----
18 12	9	Directory block two	18,17
18 12	10 $0a	----- unused -----	-----
18 12	11 $0b	----- unused -----	-----
18 12	12 $0c	----- unused -----	-----
18 12	13 $0d	----- unused -----	-----
18 12	14 $0e	----- unused -----	-----
18 12	15 $0f	----- unused -----	-----
18 12	16 $10	----- unused -----	-----
18 12	17 $11	Directory block three	18, 6
18 12	18 $12	----- unused -----	-----

Track	Sector	Contents	Link
19 13	0	Backup, block 1 (no side sector - BASIC & ML)	19,14
19 13	1	Geopaint #7, block 4 (114)	19, 9
19 13	2	Geopaint #6, block 3 (107)	19,10
19 13	3	Backup, block 3	20,14
19 13	4	Geopaint #7, block 2 (112)	19,12
19 13	5	Geopaint #6, block 1 (105)	19,13
19 13	6	Geopaint #7, block 7 (117-last block of file)	-----
19 13	7	Geopaint #6, block 6 (110-last block of part 6)	-----
19 13	8	The Border file directory block	-----
19 13	9	Geopaint #7, block 5 (115)	19,17
19 13	10 $0a	Geopaint #6, block 4 (108)	19,18
19 13	11 $0b	Backup INFO BLOCK	-----
19 13	12 $0c	Geopaint #7, block 3 (113)	19, 1
19 13	13 $0d	Geopaint #6, block 2 (106)	19, 2
19 13	14 $0e	Backup, block 2	19, 3
19 13	15 $0f	Geopaint #7, block 1 (111)	19, 4
19 13	16 $10	Geopaint #5, block 11 (104-last block of part 5)	-----
19 13	17 $11	Geopaint #7, block 6 (116)	19, 6
19 13	18 $12	Geopaint #6, block 5 (109)	19, 7
20 14	0	Backup, block 7	20, 8
20 14	1	Preference Mgr, block 1 (no side sector)	-----
20 14	2	Backup, block 12	20,10
20 14	3	Backup, block 5	20,11
20 14	4	Calculator, block 1 (no side sector)	-----
20 14	5	Backup, block 10	20,13
20 14	6	Preference Mgr, block 4	21,14
20 14	7	Calculator, block 2	21,17
20 14	8	Backup, block 8	20,16
20 14	9	Preference Mgr, block 2	20,17
20 14	10 $0a	Backup, block 13	20,18
20 14	11 $0b	Backup, block 6	20, 0
20 14	12 $0c	Preference Mgr INFO BLOCK	-----
20 14	13 $0d	Backup, block 11	20, 2
20 14	14 $0e	Backup, block 4	20, 3
20 14	15 $0f	Calculator INFO BLOCK	-----
20 14	16 $10	Backup, block 9	20, 5
20 14	17 $11	Preference Mgr, block 3	20, 6
20 14	18 $12	Backup, block 14	23, 5

Track	Sector	Contents	Link
21 15	0	Preference Mgr, block 8	21, 8
21 15	1	Preference Mgr, block 20	21, 9
21 15	2	Preference Mgr, block 13	21,10
21 15	3	Preference Mgr, block 6	21,11
21 15	4	Preference Mgr, block 18	21,12
21 15	5	Preference Mgr, block 11	21,13
21 15	6	Calculator, block 4	22,14
21 15	7	Preference Mgr, block 16	21,15
21 15	8	Preference Mgr, block 9	21,16
21 15	9	Preference Mgr, block 21 (last block of file)	-----
21 15	10 $0a	Preference Mgr, block 14	21,18
21 15	11 $0b	Preference Mgr, block 7	21, 0
21 15	12 $0c	Preference Mgr, block 19	21, 1
21 15	13 $0d	Preference Mgr, block 12	21, 2
21 15	14 $0e	Preference Mgr, block 5	21, 3
21 15	15 $0f	Preference Mgr, block 17	21, 4
21 15	16 $10	Preference Mgr, block 10	21, 5
21 15	17 $11	Calculator, block 3	21, 6
21 15	18 $12	Preference Mgr, block 15	21, 7
22 16	0	Calculator, block 8	22, 8
22 16	1	Alarm Clock, block 5	22, 9
22 16	2	Calculator, block 13	22,10
22 16	3	Calculator, block 6	22,11
22 16	4	Alarm Clock, block 3	22,12
22 16	5	Calculator, block 11	22,13
22 16	6	Alarm Clock, block 1 (no side sector)	22,15
22 16	7	Alarm Clock, block 7	23,14
22 16	8	Calculator, block 9	22,16
22 16	9	Alarm Clock, block 6	22, 7
22 16	10 $0a	Calculator, block 14	22,18
22 16	11 $0b	Calculator, block 7	22, 0
22 16	12 $0c	Alarm Clock, block 4	22, 1
22 16	13 $0d	Calculator, block 12	22, 2
22 16	14 $0e	Calculator, block 5	22, 3
22 16	15 $0f	Alarm Clock, block 2	22, 4
22 16	16 $10	Calculator, block 10	22, 5
22 16	17 $11	Alarm Clock INFO BLOCK	-----
22 16	18 $12	Calculator, block 15 (last block of file)	-----

Track	Sector	Contents	Link
23 17	0	Alarm Clock, block 11	23, 8
23 17	1	Note Pad, block 5	23, 9
23 17	2	Note Pad, block 11	23,13
23 17	3	Alarm Clock, block 9	23,11
23 17	4	Note Pad, block 3	23,12
23 17	5	Backup, block 15	35, 4
23 17	6	Note Pad, block 8	23,16
23 17	7	Note Pad, block 1 (no side sector)	23,15
23 17	8	Alarm Clock, block 12 (last block of file)	-----
23 17	9	Note Pad, block 6	23,17
23 17	10 $0a	Note Pad, block 10	23, 2
23 17	11 $0b	Alarm Clock, block 10	23, 0
23 17	12 $0c	Note Pad, block 4	23, 1
23 17	13 $0d	Note Pad, block 12	24,14
23 17	14 $0e	Alarm Clock, block 8	23, 3
23 17	15 $0f	Note Pad, block 2	23, 4
23 17	16 $10	Note Pad, block 9	23,10
23 17	17 $11	Note Pad, block 7	23, 6
23 17	18 $12	Note Pad INFO BLOCK	-----
24 18	0	Note Pad, block 16 (last block of file)	-----
24 18	1	Photo Manager INFO BLOCK	-----
24 18	2	Photo Manager, block 7	24,10
24 18	3	Note Pad, block 14	24,11
24 18	4	Photo Manager, block 5	24,12
24 18	5	Photo Manager, block 12	24,13
24 18	6	Photo Manager, block 3	24,15
24 18	7	Photo Manager, block 10	24,16
24 18	8	Photo Manager, block 14	25,14
24 18	9	Photo Manager, block 1 (no side sector)	24,17
24 18	10 $0a	Photo Manager, block 8	24,18
24 18	11 $0b	Note Pad, block 15	24, 0
24 18	12 $0c	Photo Manager, block 6	24, 2
24 18	13 $0d	Photo Manager, block 13	24, 8
24 18	14 $0e	Note Pad, block 13	24, 3
24 18	15 $0f	Photo Manager, block 4	24, 4
24 18	16 $10	Photo Manager, block 11	24, 5
24 18	17 $11	Photo Manager, block 2	24, 6
24 18	18 $12	Photo Manager, block 9	24, 7

Track	Sector	Contents	Link
25 19	0	Text Manager, block 4	25, 8
25 19	1	Text Manager, block 6	25,12
25 19	2	Photo Manager, block 21	25, 9
25 19	3	Photo Manager, block 16	25,10
25 19	4	Text Manager, block 2	25,11
25 19	5	Text Manager, block 8	26,14
25 19	6	Photo Manager, block 19	25,13
25 19	7	Text Manager INFO BLOCK	-----
25 19	8	Text Manager, block 5	25, 1
25 19	9	Photo Manager, block 22 (last block of file)	-----
25 19	10 $0a	Photo Manager, block 17	25,17
25 19	11 $0b	Text Manager, block 3	25, 0
25 19	12 $0c	Text Manager, block 7	25, 5
25 19	13 $0d	Photo Manager, block 20	25, 2
25 19	14 $0e	Photo Manager, block 15	25, 3
25 19	15 $0f	Text Manager, block 1	25, 4
25 19	16 $10	BlueChip M120 INFO BLOCK	-----
25 19	17 $11	Photo Manager, block 18	25, 6
26 1a	0	California Font SIDE SECTOR: 26,15 (4 blocks) 27,10 (5 blocks) 27,11 (6 blocks) 27,16 (9 blocks)	-----
26 1a	1	BlueChip M120, block 1	26, 8
26 1a	2	Text Manager, block 15	26, 9
26 1a	3	Text Manager, block 10	26,10
26 1a	4	California #1, block 2	26, 7
26 1a	5	Text Manager, block 18	26,12
26 1a	6	Text Manager, block 13	26,13
26 1a	7	California #1, block 3	27,14
26 1a	8	BlueChip M120, block 2 (last block of file)	26, 1
26 1a	9	Text Manager, block 16	26,16
26 1a	10 $0a	Text Manager, block 11	26,17
26 1a	11 $0b	California Font INFO BLOCK	-----
26 1a	12 $0c	Text Manager, block 19 (last block of file)	-----
26 1a	13 $0d	Text Manager, block 14	26, 2
26 1a	14 $0e	Text Manager, block 9	26, 3
26 1a	15 $0f	California #1, block 1	26, 4
26 1a	16 $10	Text Manager, block 17	26, 5
26 1a	17 $11	Text Manager, block 12	26, 6

Track	Sector	Contents	Link
27 1b	0	California #3, block 2 (11)	27, 7
27 1b	1	California #4, block 3 (18)	27, 8
27 1b	2	California #2, block 5 (9-last block of part 2)	-----
27 1b	3	California #4, block 5 (20)	27, 9
27 1b	4	California #3, block 5 (14)	27,12
27 1b	5	California #4, block 2 (17)	27, 1
27 1b	6	California #2, block 3 (7)	27,13
27 1b	7	California #3, block 3 (12)	27,15
27 1b	8	California #4, block 4 (19)	27, 3
27 1b	9	California #4, block 6 (21)	28,14
27 1b	10 $0a	California #2, block 1 (5)	27,17
27 1b	11 $0b	California #3, block 1 (10)	27, 0
27 1b	12 $0c	California #3, block 6 (15-last block of part 3)	-----
27 1b	13 $0d	California #2, block 4 (8)	27, 2
27 1b	14 $0e	California #1, block 4 (last block of part 1)	-----
27 1b	15 $0f	California #3, block 4 (13)	27, 4
27 1b	16 $10	California #4, block 1 (16)	27, 5
27 1b	17 $11	California #2, block 2 (6)	27, 6
28 1c	0	Cory #2, block 4 (9)	28, 7
28 1c	1	Cory Font SIDE SECTOR: 28,13 (5 blocks) 28,15 (16 blocks)	-----
28 1c	2	Cory #1, block 2	28, 9
28 1c	3	California #4, block 8 (23)	28,10
28 1c	4	Cory #2, block 2 (7)	28,11
28 1c	5	Cory #1, block 5 (last block of part 1)	-----
28 1c	6	Cory #2, block 7 (12)	28, 8
28 1c	7	Cory #2, block 5 (10)	28,17
28 1c	8	Cory #2, block 8 (13)	29,14
28 1c	9	Cory #1, block 3	28,16
28 1c	10 $0a	California #4, block 9 (24-last block of file)	-----
28 1c	11 $0b	Cory #2, block 3 (8)	28, 0
28 1c	12 $0c	Cory Font INFO BLOCK	-----
28 1c	13 $0d	Cory #1, block 1	28, 2
28 1c	14 $0e	California #4, block 7 (22)	28, 3
28 1c	15 $0f	Cory #2, block 1	28, 4
28 1c	16 $10	Cory #1, block 4	28, 5
28 1c	17 $11	Cory #2, block 6 (11)	28, 6

Track	Sector	Contents	Link
29 1d	0	Dwinelle, block 6	29, 7
29 1d	1	Dwinelle, block 4	29, 8
29 1d	2	Cory #2, block 15 (20)	29, 9
29 1d	3	Cory #2, block 10 (15)	29,10
29 1d	4	Dwinelle Font SIDE SECTOR: 29,16 (11 blocks)	-----
29 1d	5	Dwinelle, block 2	29,12
29 1d	6	Cory #2, block 13 (18)	29,13
29 1d	7	Dwinelle, block 7	29,11
29 1d	8	Dwinelle, block 5	29, 0
29 1d	9	Cory #2, block 16 (21-last block of file)	-----
29 1d	10 $0a	Cory #2, block 11 (16)	29,17
29 1d	11 $0b	Dwinelle, block 8	30,14
29 1d	12 $0c	Dwinelle, block 3	29, 1
29 1d	13 $0d	Cory #2, block 14 (19)	29, 2
29 1d	14 $0e	Cory #2, block 9 (14)	29, 3
29 1d	15 $0f	Dwinelle INFO BLOCK	-----
29 1d	16 $10	Dwinelle, block 1	29, 5
29 1d	17 $11	Cory #2, block 12 (17)	29, 6
30 1e	0	Roma #2, block 1 (5)	30, 7
30 1e	1	Roma #3, block 1 (10)	30, 8
30 1e	2	Roma #1, block 4 (last block of part 1)	-----
30 1e	3	Dwinelle, block 10	30,10
30 1e	4	Roma #2, block 4 (8)	30,11
30 1e	5	Roma Font SIDE SECTOR: 30,17 (4 blocks) 30, 0 (5 blocks) 30, 1 (9 blocks) 30,5 (14 blocks)	-----
30 1e	6	Roma #1, block 2	30,13
30 1e	7	Roma #2, block 2 (6)	30,15
30 1e	8	Roma #3, block 2 (11)	30, 9
30 1e	9	Roma #3, block 3 (12)	30,12
30 1e	10 $0a	Dwinelle, block 11 (last block of file)	-----
30 1e	11 $0b	Roma #2, block 5 (9-last block of part 2)	-----
30 1e	12 $0c	Roma #3, block 4 (13)	31,14
30 1e	13 $0d	Roma #1, block 3	30, 2
30 1e	14 $0e	Dwinelle, block 9	30, 3
30 1e	15 $0f	Roma #2, block 3 (7)	30, 4
30 1e	16 $10	Roma Font INFO BLOCK	-----
30 1e	17 $11	Roma #1, block 1	30, 6

Track	Sector	Contents	Link
31 1f	0	Roma #4, block 10 (28)	31, 7
31 1f	1	Roma #3, block 8 (17)	31, 8
31 1f	2	Roma #4, block 3 (21)	31, 9
31 1f	3	Roma #4, block 8 (26)	31,10
31 1f	4	Roma #3, block 6 (15)	31,11
31 1f	5	Roma #4, block 1 (19)	31,12
31 1f	6	Roma #4, block 6 (24)	31,13
31 1f	7	Roma #4, block 11 (29)	31,15
31 1f	8	Roma #3, block 9 (18-last block of part 3)	-----
31 1f	9	Roma #4, block 4 (22)	31,16
31 1f	10 $0a	Roma #4, block 9 (27)	31, 0
31 1f	11 $0b	Roma #3, block 7 (16)	31, 1
31 1f	12 $0c	Roma #4, block 2 (20)	31, 2
31 1f	13 $0d	Roma #4, block 7 (25)	31, 3
31 1f	14 $0e	Roma #3, block 5 (14)	31, 4
31 1f	15 $0f	Roma #4, block 12 (30)	32,14
31 1f	16 $10	Roma #4, block 5 (23)	31, 6
32 20	0	University #2, block 4 (6-last block of part 2)	-----
32 20	1	University #3, block 2 (8)	32,11
32 20	2	University #3, block 4 (10)	32,12
32 20	3	University #4, block 1 (12)	32, 5
32 20	4	Roma #4, block 14 (32) (last block of file)	-----
32 20	5	University #4, block 2 (13)	33,14
32 20	6	University Font INFO BLOCK	-----
32 20	7	University #1, block 1	32,15
32 20	8	University #2, block 1 (3)	32,16
32 20	9	University #2, block 3 (5)	32, 0
32 20	10 $0a	University #3, block 1 (7)	32, 1
32 20	11 $0b	University #3, block 3 (9)	32, 2
32 20	12 $0c	University #3, block 5 (11-last block of part 3)	-----
32 20	13 $0d	University Font SIDE SECTOR:	-----
		32,7 (2 blocks	
		32,8 (4 blocks)	
		32,10 (5 blocks)	
		32,3 (6 blocks)	
		33,8 (8 blocks	
		33,10 (13 blocks)	
32 20	14 $0e	Roma #4, block 13 (31)	32, 4
32 20	15 $0f	University #1, block 2 (last block of part 1)	-----
32 20	16 $10	University #2, block 2 (4)	32, 9

Track	Sector	Contents	Link
33 21	0	University #6, block 2 (27)	33, 7
33 21	1	University #4, block 6 (17-last block of part 4)	-----
33 21	2	University #5, block 5 (22)	33, 9
33 21	3	University #6, block 4 (29)	33,13
33 21	4	University #4, block 4 (15)	33,11
33 21	5	University #5, block 3 (20)	33,12
33 21	6	University #5, block 8 (25-last block of part 5)	-----
33 21	7	University #6, block 3 (28)	33, 3
33 21	8	University #5, block 1 (18)	33,15
33 21	9	University #5, block 6 (23)	33,16
33 21	10 $0a	University #6, block 1 (26)	33, 0
33 21	11 $0b	University #4, block 5 (16)	33, 1
33 21	12 $0c	University #5, block 4 (21)	33, 2
33 21	13 $0d	University #6, block 5 (30)	34,14
33 21	14 $0e	University #4, block 3 (14)	33, 4
33 21	15 $0f	University #5, block 2 (19)	33, 5
33 21	16 $10	University #5, block 7 (24)	33, 6
34 22	0	C.Itoh 8510, block 1 (no side sector)	34, 7
34 22	1	University #6, block 9 (34)	34, 8
34 22	2	Epson FX-80 INFO BLOCK	-----
34 22	3	MPS-801, block 1	34,16
34 22	4	University #6, block 7 (32)	34,11
34 22	5	University #6, block 12 (37)	34,12
34 22	6	MPS-801, block 3 (last block of file)	-----
34 22	7	C.Itoh 8510, block 2 (last block of file)	-----
34 22	8	University #6, block 10 (35)	34,15
34 22	9	Epson FX-80, block 1 (no side sector)	35,14
34 22	10 $0a	C.Itoh 8510 INFO BLOCK	-----
34 22	11 $0b	University #6, block 8 (33)	34, 1
34 22	12 $0c	University #6, block 13 (38-last block of file)	-----
34 22	13 $0d	MPS-801 INFO BLOCK	-----
34 22	14 $0e	University #6, block 6 (31)	34, 4
34 22	15 $0f	University #6, block 11 (36)	34, 5
34 22	16 $10	MPS-801, block 2	34, 6

Track	Sector	Contents		Link
35 23	0	Comm.Compat. INFO BLOCK		-----
35 23	1	Comm.Compat., block 2		35, 8
35 23	2	Epson MX-80, block 1	(no side sector)	35, 9
35 23	3	Joystick, block 1	(no side sector)	35,10
35 23	4	Backup, block 16	(last block of file)	-----
35 23	5	MPS-1000, block 1	(no side sector)	35,16
35 23	6	MPS-1000, block 3	(last block of file)	-----
35 23	7	Comm.Compat., block 1	(no side sector)	35, 1
35 23	8	Comm.Compat., block 3	(last block of file)	-----
35 23	9	Epson MX-80, block 2	(last block of file)	-----
35 23	10 $0a	Joystick, block 2	(last block of file)	-----
35 23	11 $0b	----- unused -----		-----
35 23	12 $0c	Epson MX-80 INFO BLOCK		-----
35 23	13 $0d	Joystick INFO BLOCK		-----
35 23	14 $0e	Epson FX-80, block 2	(last block of file)	-----
35 23	15 $0f	MPS-1000 INFO BLOCK		-----
35 23	16 $10	MPS-1000, block 2		35, 6

GEOS in Memory

Any attempt to describe GEOS inside the computer's memory is foredoomed to failure. As a disk based operating system, GEOS swaps information back and forth between the disk drive and the computer at need. This is especially evident in the case of applications. Consider the following example.

You are working on a geoWrite document. You start off the page with an attractive banner in the Dwinelle 18 point font, then continue with Roma 12 point. You have prepared a few pieces of text and several graphics for use in this particular document. When you get to the appropriate spot, you pull in the Text Scrap that holds a quote from another document. A bit later, you want to use a graph that you created with geoPaint, so you pull in the Photo Scrap. On page 2, now, you need another quote that you have stored in an album on disk, so you select GEOS/text manager, put the quote in the Text Scrap, and pull the Scrap into your document. Finally, you add a digitized picture of yourself that went through Doodle format, then imported into geoPaint, and from there to an album. You pull this photo in and place a caption underneath it in University 6 point.

In this session with geoWrite, you have twice loaded a desk accessory. When you did this, part of geoWrite was saved to disk, then called back into the computer when you closed the desk accessory. You have two pages of text, with only one in the computer at any one time, and two graphics which are stored separately within the computer. The memory management required by all this is impressive.

Parts of GEOS seem to remain resident in the computer at all times; other areas are consistently used for primary and secondary screens, programs, and program data.

Again, the key to finding where data sits in memory is the disk. The following pages give a map of the information given in the disk directory entries and the information block. The load addresses given for some programs in bytes 71 and 72 are valid.

Length Blocks	Desktop Page	Filename	Load Addr.	C= pointer to first block or Side Sector	Type of data there	Geos info block pntr.	C= prg type	Geos file type #	Sprite
1	p.1	Geos	$0110	1,8	code	--,--	prg<	--	64 logo
6	p.1	Geos Boot	$5ffe	1,17	code	1,9	prg<	02	boot
85	p.1	Geos Kernal	-----	1,18	side sector:	1,10	usr<	14	globe
17			$9000		1,11				
66			$bf40		2,15				
72	p.1	Desk Top	$0406	5,8	side sector:	5,0	usr<	14	desk
62			$0406		5,16				
8			$5000		8,2				
119	p.1	Geo Paint	$0400	8,19	side sector:	8,11	usr<	16	pallete
64					8,3				
7					16,17				
10					16,10				
12					16,6				
11					17,18				
6					19,5				
7					19,15				
88	p.1	Geo Write	$0400	9,14	side sector:	8,6	usr<	16	quill
36			$0400		9,1				
14			$3438		10,16				
10					11,2				
7					11,19				
7					12,12				
7					12,5				
5					12,19				
17	p.1	Backup	$07ff	19,0	code (Basic and ML)	19,11	usr<	01	disks
22	p.2	Prefrence Mgr	$4b00	20,1	code	20,12	usr<	05	station
13	p.2	Alarm Clock	$5400	22,6	code	22,17	usr<	05	alarm clock
23	p.2	Photo Manager	$2e00	24,9	code	24,1	usr<	05	two windows
20	p.2	Text Manager	$2e00	25,15	code	25,7	usr<	05	two windows
16	p.2	Calculator	$5200	20,4	code	20,15	usr<	05	calculator
17	p.2	Note Pad	$5000	23,7	code	23,18	usr<	05	note pad

				C= pointer to				Geos	
Length	Desktop		Load	first block or		Geos info	C= prg	file	
Blocks	Page	Filename	Addr.	Side Sector	Type of data there	block pntr.	type	type #	Sprite
26	p.3	California	-----	26,0	side sector:	26,11	usr<	18	font in box
4					26,15				
5					27,10				
6					27,11				
9					27,16				
23	p.3	Cory	-----	28,1	side sector:	28,12	usr<	18	font in box
5					28,13				
16					28,15				
13	p.3	Dwinelle	-----	29,4	side sector:	29,15	usr<	18	font in box
11					29,16				
34	p.3	Roma	-----	30,5	side sector:	30,16	usr<	18	font in box
4					30,17				
5					30,0				
9					30,1				
14					31,5				
40	p.3	University	-----	32,13	side sector:	32,6	usr<	18	font in box
2					32,7				
4					32,8				
5					32,10				
6					32,3				
8					33,8				
13					33,10				
4	p.4	MPS-801	$7900	34,3	code	34,13	usr<	09	printer
4	p.4	MPS-1000	$7900	35,5	code	35,15	usr<	09	printer
4	p.4	Comm. Compat.	$7900	35,7	code	35,0	usr<	09	printer
3	p.4	BlueChip M120	$7900	26,1	code	25,16	usr<	09	printer
3	p.4	Epson FX-80	$7900	34,9	code	34,2	usr<	09	printer
3	p.4	Epson MX-80	$7900	35,2	code	35,12	usr<	09	printer
3	p.4	C.Itoh 8510	$7900	34,0	code	34,10	usr<	09	printer
3	p.4	Joystick	$fe80	35,3	code	35,13	prg<	0a	joystick

BOOTING GEOS

The file "GEOS" on disk may be LOADed and RUN, or LOADed ,8,1. If LOADed and RUN, the BASIC program LOADs "GEOS",8,1. You end up with the same thing and the BASIC program with a machine language tail will load into memory at $0110 (+272), across the processor stack area and into the BASIC input buffer. This automatically executes, loads "GEOS BOOT" into memory at $6000, and JMPs to $6000.

"GEOS BOOT" includes about one Kbyte of encrypted code. If you load "GEOS BOOT" directly into memory, you can see that most of the code after $6088 looks like garbage. As loaded, it is.

The beginning of the "GEOS BOOT" routine at $6000 immediately JMPs to $606c. This routine decrypts the code following $6088. Following the code from $606c on through $6088, you can see how this routine sets bits zero, one, three, and four in all bytes in memory from $6089 through $6489. To look at this entire "GEOS BOOT," insert a BRK at $6089 and JMP or SYS to 606c. The code may then be examined.

The "GEOS BOOT" file also contains much code that is sent back to the disk drive as a fast load routine. Another fast disk routine is loaded later with the "GEOS KERNAL" which is used after GEOS is up and running.

A brief map of the "GEOS BOOT" file is given on the following pages. It is not totally disassembled here, and not all locations are explained, but the major routines and registers are noted. The code for the drive will at regular intervals show two addresses. The addresses in the $0300 - $04ee range are the correct addresses within the drive.

The explantions given here pertain to GEOS V1.2 as shipped during the summer and fall of 1986. Later revisions may be different. The code of "GEOS BOOT" has been decrypted by inserting a BRK at $6086, then JMPing to $606c. The byte at $62cc was changed from a $1d to a $a2 before decryption.

```
6000                JMP to decrypt routine at 606c
6003                text for M-E?
6013                lands from JSRs at $60a3 and $60aa
                    fast load?
6016 - 601d         wait for reply
601e - 6023         delay loop
--
6065                land from JSR at $61c1
--
606c                entry point for boot routine
--
607a                changed by $61da to $a9 (LDA #$27)
--
6087                changed by $61e4 to JMP $fce2 (power on reset)
6089 - 6098         "BOOTING GEOS..." in screen code
                    referenced by 616e
6099                land from JSRs at $61d1 and $61fa
--
60b0 - 60b3         dump BASIC and KERNAL out
60b5 - 60ba         get code from low memory ($623f) and
                    store it in high memory ($9000)
                    see $61c9 - $61d0.
60bc                bring KERNAL back
--
60cd - 60d1         "M-E" text pointed to by $61af
60d2 - 60d6         "M-W" text pointed to by $60d7
                    modified by 617f to write to drive buffer at
                    $0300.
60d7 - 60f5         set up for M-W to disk
                    called by JSR at $618d
60f6 -              land from JMP at $6086
                    dump BASIC ROM, reset timers, turn off
                    sprites, set interrupts
6118 -              set IRQ vector to $6239
6120 -              set NMI vector to $623e
614f -              blank screen
6167 -              set colors and print "BOOTING GEOS..."
6177 -              send fast load routine to drive at drive
                    addresses $0300 through $04ee?
61b5 -              set up M-E for drive address $0375
```

```
--             wait for drive
61d1 -         fast load first half of "GEOS KERNAL"
61fa -         fast load second half of "GEOS KERNAL"
621f -         JMP to KERNAL code at $c003
--
6222 -         land from JSR at $60df and $61b5
6224 -         command device to listen
6229 -         secondary address after listen?
6230 -         output byte to serial bus
6239 -         IRQ routine
623e -         NMI sent here
-----------------------------------------------------------

DRIVE CODE
0300 623f      same as  $6003  to  $6012:  channel,  secondary
               address, and "ME"
--
0310 624f      land from JSR at $04dc
--
0327 6266      land from JMP at $04ea and JSR at $031c
0362 62a1      set data line out
0368 62a7      land from JSR at  $04e4, $04dc,  $0317, $0321: set
               clock in line
0375 62b4      M-E comes here to start drive code
--
     62cc      byte altered by Easy Load utility.
--
0492 63d1      sector of GEOS KERNAL side sector
     63d7      track of GEOS KERNAL side sector
```

ZERO PAGE

0	DDR ($2f)
1	I/O assignments ($30, indicating KERNAL and BASIC out...Snapshot changes this to $35?
4-5	often used as track # and sector # storage. - also used as an address pointer by J.T. routine via $c181 (JMP table to $ce73)
0c-0d	used as pointer to dialog box parameters by $f3d8-$f3e1
2-25	addresses often used for indirect indexing
26-2d	font information, values come from $d210 for default font
2a-2d	values initially stored here by $d210, later modified by $dfd4-$dfed
31-32	points to the arrow sprite data at $84c1
30, 39-3	contain input driver information
41-2	used to enter routines by jumpt table $c1d8, other routines have tables of address placed into this location for vector loads
73-8a	charget remains unchanged
d9-f2	line link table remains unchanged
fd-fe	used by deskTop at $0882 and $0889
0110-0205	'geos' loads into stack at this location

0203 is the JMP within the input buffer that activates the 'geos' programs stack activated jump within the input buffer when loaded ,8,1

02fe-ff geos boot stores current raster screen lines here

VECTORS

0300-0333, all values standard, except:

0314-0315 IRQ vector pointing to $c542

0318-0319 NMI vecter pointing to $c547 Upon exit to BASIC, only the NMI is left altered, in which case it points to $c000 which will reboot GEOS if the RESTORE key is pressed.

0400 geoPaint and geoWrite load here

0406 desktop load address

2e00 photo and text manager load here, swapping code into deskTop

2e6c-2e6f "RUN"

2ec1-2ecc "SYS"

2f12-2f40 "LOAD"

3349 copies BAM from $8200 to $4a00

381c-3f30 deskTop text messages reside here

40fc end of deskTop

40fd - 49ff deskTop workspace

4102 deskTop Notepad page number. 0=1, 1=2, 2=3, 3=4, etc.

4105 number of files on disk

411c-412b name of first file on active deskTop Notepad page

4144 name of input driver

4197-41a8 disk name

41a9-41ba disk name

41cc-427c table of values???

4281-4290 pointers to text locations in the appropriate directory block for the current deskTop notepad page you are on. The directory blocks are loaded in at $4b00-$4eff. Used by $31a0-$31ad

42b5-42d2 complete directory entry of file for which File/info was requested.

42de Info block can appear here

4328 geoWrite document page text storage starts here, in ASCII

437e Info block text goes here, can be added to.

4970 buffer for storage of desk accessory names available on this disk, used by GEOS/* menu

4a00 - copy of BAM transferred down from $8200 by desktop at $3349

4b00 - directory block 1, 18/1
4c00 - directory block 2, 18/9
4d00 - directory block 3, 18/17
4e00 - directory block 4, 18/6
4eff end of actual directory information
4f00 - space for directory block

5000-5723 second part of deskTop loads in here.

5000 notepad loads here

5200 calculator loads here

5400 alarm clock loads here

4b00-5fc8 preferences manager loads here

5c00 border directory block

SPRITE DATA STORAGE

5d00-60ff beginning of sprite data - the first $80 (128) bytes of information blocks, including the $00, $ff track/sector pointers, for disk files are pulled into memory starting at this location. Actual data starts at $5d80

5e68 load address of all fonts, all point sizes.

5f67-5fae "Preferences" file loads in here, right near the end of the Preference Manager.

6000 - 7f3f secondary screen/background screen saved screen

6000-642b GEOS BOOT temporarily resides here

7900 printer driver load address

7f40-7fff geoPaint uses this area

8000 last block read in. working drive buffer.
8100 INFO block of file currently being loaded in

8200 Primary BAM storage, copied from here to $4a00 by desktop at $3349

8300 file track and sector link table - record of which track and sector each loaded block came from; sequential list of the track and sectors of the blocks that have been loaded in.

8400-8487 work area for building directories or directory entry of a file being used. Pointed to by $c978-7f, using $14-5. default input driver name and disk name, preferences filename, if file Preferences on disk.
8400 Commodore DOS filetype 8401-8402 track/sector link
8403-8412 filename
8413-8414 information block track/sector link
8415 GEOS file structure flag
8416 GEOS file type byte 1 loaded by $c980
8417-841b date/time YY/MM/DD/hh/mm
841c-841d file size in 254 byte blocks

841e-842f standard name of disk in device 8
8430-8441 standard name of disk in device 9
8442-8452 name of application file (geoW or P document)
8453-8464 name of disk that the application data is on
8465-8475 default printer driver name
8476-8487 disk name default printer driver pulled from
8489 current device number pulled from $ba several times in $c000

848b is it a GEOS disk flag. $ff=yes, $00=no. see $cb93 - $cbbl

848c used in BAM manipulation at $9661, $964a; set to #$08 by $cc02 initially

848d number of drives

849b-84b2 vectors for indirect jumps [JMP(addr)]

849b-c vector into deskTop

84af-84b0 break (BRK) vector

84b3-84c0	flags?	
84c1-84ff	arrow sprite data	
8501-3	joystick or mouse values	
8505-7	joystick or mouse values	
850c-8513	font information, same values as stored at $26-$2d	
8516	year	put there by $cfdc
8517	month	$cfda through $cfe3
8518	day	put here by $cfe4 through
8519	hour	$cfed
851a	minutes	stored with $00 by $cfd4
851b	seconds	stored with $00 by $cfd7 (TOD)
851c	stored with $00 at $cff0	
851d	Dialogue box index number. stored with $00 by $f284, picked up by routine at $f47a. Dialogue box index number.	
851e	stored with $bf by $cdfe. default screen color	
87da-87e9	keypress input buffer used by geoWrite, calculator, etc.	
8850-8851	PC high/PC low storage	
8852	stack pointer placed here by $91b8, $9182	
8853-8854	PC high/PC low storage	
8855	stack pointer placed here by $f239	
885e	processor status flags came from $c4c3	
885f	irq mask stored here by $c4d0	
8860	value from location $01 by $c4c9, restored by $c560	
8861	sprite enable register by $c516	
8865	temporary track storage	
8866	temporary sector storage	

8867 serial information of $dd00 (data port a) by $c533 the remaining is similar storage

886e used to flag read through last directory block at $c560

8986-898d CBM drive error message buffer, sometimes

8a00 sprite #0 data - arrow sprite usually
8a40 sprite #1 data
8a80 sprite #2 data
8ac0 sprite #3 data
8b00 sprite #4 data
8b40 sprite #5 data
8b80 sprite #6 data
8bc0 sprite #7 data

8c00-8fe7 hires color memory $BFs placed here by $cded and its JSR to $ce51

8fe8-8ff7 16 $00s unused

8ff8-8fff sprite pointers 0 through 7 always $28,$29,$2a,$2b,$2c,$2d,$2e,$2f

GEOS KERNAL #1 ($9000-9fff)

9000-902f OPEN a disk

9030-9066

9067-90e4 find file of a given type. $10=file type, $0e-f= pointer to filename storage buffer and upon exit pointer to track and sector of found file, $11=high byte of number of bytes to be zeroed and file found flag.

90e5-911a loads INFO block of found file.

911b-918e routine used to SWAP in desk accessories using file entry point data from the INFO block which only desk accessories, deskTop, applications, GEOS kernal and GEOS boot have. Jump table referenced at $c217. Pointed to by table at $c98f-$c9a6 using ($41).

912e-9132 points to swap file text at $989c

9191-9195 points to swap file text at $98c9

91e7-9203 set a pointer to the current disk name for the currently active drive

924b-9268 called by $92ee and from deskTop at $17be, and ten times from "CONVERT". Send a block with the preparation work being done such as sending FAST I/O code to the drive, and starting it up, setting up I/Os, using $9269-$92a3 to send the block, and closing down I/Os when it is done

9269-92a3 called by $9254 and $92e0. Actual code that sends code to the disk

92a4-92ea called by $9266, $925a. If an error condition occurs in sending a block to the drive, this routine tries it again.

92f1-93a7 SAVE. Jump table referenced at c1ed. Called by $9898 via jump table.

935c-9382 calculate file length and leave total in ($06-7).

9532-95bd transfers saved file data. Jump table referenced at $c1f9. Called by $9358 direct.

96b9-96bd Table of sectors per track. This table appears in deskTop at $1b4b-$1b69 also.

96dc-9701 BAM level routine
9702-9720 BAM level routine

9844-985b BAM level routine
985c-987c Calculate the number of blocks free from BAM in RAM at $8200-$82ff and leave total in ($0a-b).

989c-98 swap file text. referred to by $912e-$9132 and $9191-$9195

9c79-9ca6 storage area for routine below (i.e. $9ca7)
9ca7-9d92 go to BASIC, called by JMP at $c241 (see c241).
9d93-9de6 NMI routine associated with $9ca7-$9d92

9e0d-9e4e position a sprite. called directly by $e19a ; and via the table ($c1cf) by $e615

9e4f-9e66 turn sprite on using enable register at $d015, called directly by $e19d, and via table ($c1d2) by $e61c and from deskTop at $21e0

9d93-9de6 set up NMIs for returning to BASIC

9de7-9dfc store sprite data (see $c1c6)

9dfd-9e04 lo-byte table for sprite storage addresses

9e05-9e0c hi-byte table for sprite storage addresses

A000-bf3f MAIN HIRES SCREEN. Active, primary screen

GEOS KERNAL #2 $bf40-$fff9

bf40-bf57 significant data for arrow sprite (up to the trailing zeros, two inlcuded). Written to zeroed out area at $8a00

bf58-bfaa 'CANCEL' in a box (for dialogue boxes)

bfac-bfff OK' in a dialogue box

C000 reboots GEOS - JMP $c010 as long as you preserve everything from $c000 to $c037, you can reboot GEOS

c003 JMP $cc21 initialization routine following GEOS BOOT

c00f GEOS version identifier byte. $12 indicates version 1.2. see "CONVERT" ($0b36-$0b3c)

c038-c056 table of fast I/O data register values

c057-c0a5 computer side receiver of fast I/O.
c057-c0a5 routine for computer to receive information from the computer fast

c09f-c0ff computer side sender of fast I/O... routine for computer to send information to the drive fast

c100-c2c7 jump table 152 total, 151 active jumps, #148 NOPed out

c181 JMP $ce73: Write Values to RAM. $02,$03 point to the data. Called by $e606 and $21bd. Byte 1 of the data indicates target address low; byte 2 target address high, byte 3 # of characters to be moved

c1c6 JMP 9de7: (- 9dfc) STORE SPRITE DATA
$9dfd-$9e04 - low byte table for sprite storage address.
$9e05-$9e0c - high byte table for sprite storage address.

c1de JMP $cb93: "IS IT A GEOS DISK" called by $9014, $ca68. Upon entry into that routine, pointer set up at $0c and $0d, points to the BAM loading zone at $8100-$81ff. Upon exit, $848b and .A flags it: $ff=yes, $00= no

c1e4 JMP $c469:

cle7 JMP $924b:

clff JMP $cb00: (-$cb92.) LOAD DATA FROM DISK and construct link table. Called by $9cf8, $915c, $cace.

c208 JMP $c9c6: PROCESS GEOS file type byte.

c20b JMP $c9a7: FIND A FILE. Called by $cc65, $c96f, $cc82 and $0b94 in the public domain program, "Convert".

c214 JMP $c2fe: SEND FAST I/O to the drive and start it.

c21a JMP $c479: (-c4c0) GET A BLOCK FROM DISK. Called by $c472 and $cb26.

c229 JMP $cac7: LOAD INFO BLOCK into $8100 and set up pointers to side sector track and sector

c232 JMP $c34b: SHUT DOWN DRIVE AND SET flags

c235 JMP $c387: SHUT DOWN DRIVE AND ZERO flags

c241 JMP $9ca7: GO TO BASIC routine for selecting a standard Commodore program. goes to $9ca7-$9d92, called from desktop $2ddc, $2e57, $2e69, $2eal, $2ebd. NMI routine associated with this at $9d93-9de6. These deskTop routines contain such text as LOAD, RUN, and SYS.

c247 JMP $c957: SET TRACK AND SECTOR POINTERS, and memory storage address for loading the BAM into $8200, then does it by "JMP ($cle4)" to $c469. Called by $ca5a, $9006, $92fb, $9d0d

c256 JMP $flbb: DIALOG BOX comes from JSR at $cc90. [for example, $cc88-$cc8f loads pointer to 02 and 03 to set up dialogue box information for this routine]

c25c JMP $c4c1: FLIP IN KERNAL AND I/O, and prepare I/O for use

c25f JMP $c548: FLIP OUT KERNAL AND I/O

c265 JMP $fb82: MEMORY MOVE .A=#$00 then ($c268) $fb84.
c268 JMP $fb84: .A=# of char. (a zero will allow one byte), .X= indirect zero page load, .Y=indirect 0 page store. Called by $902a, $cf20, and $cf31 via c268. The routine at $fb84 moves a section of memory.

c26b JMP $fc44: MEMORY COMPARE .A=#$00 then ($c26e) $fc46.
c26e JMP $fc46: .A=# of char. (a zero will allow one byte), .X= indirect zero page load, .Y=indirect 0 page compare. Compares two sections of memory

c292 JMP $965e: SEARCH FOR BLOCK FREE Called by JSR at $29de in deskTop.

c298 JMP $91e7: SET POINTER TO DISK NAME from the currently active drive. Called by $1f25, $248b, $2511. Routine called directly byt $9021

c2a1 JMP $9000: OPEN A DISK. Called by $20c2 and $2d1f in deskTop. Open a disk by loading in the BAM, checking if it is a GEOS disk, and store the currently active disk name in the buffer at $841e-$8441

c2b0 JMP $c934: SET CURRENT DEVICE for the C64 and GEOS operating system, and shuts down fast I/O in drive

c2bf JMP $f32f: RETURN FROM DIALOG BOX Called by $f495, $f4a8, $f551 c2c2 JMP $fcbd: BRK ROUTINE

END OF JUMP TABLE

c2c8-c2e3 GEOS MAIN LOOP

c2e4 RTI, and that is all comes here from hardware NMI and hardware RESET from the GEOS Kernal at $fffa-$fffb, $fffc-$fffd

c2e5-c2fd called by $ccld. Zero out flags at $8492-$8495.

c2fe-c345 calls subroutine at $c397 to write code to the drive, and executes the drive code. It is referenced by the jump table at $c214. This routine is called at $e435 via its J.T. entry point. $e435 is itself a J.T. routine ($clel).

c346-c34a text for memory execute execution begins at $03dc inside the drive

c34b-c386 (c232) shut down drive and set flags

c387-c396 (c235) shut down drive and zero out flags

c397-c403 writes code to the drive in 32 byte chunks, 27 passes this is a subroutine called by $c2fe-$c345

c404-c408 text for memory write begins writing at $0300 (buffer 0)

c409-c434 subroutine for sending the M-W or M-E text and address to the drive

c435-c460 called by $9000 directly, called by other code via JMP at $clel. This routine seems to read the BAM into the drive's RAM only.

c461-c468 set pointer for storage to $8000 of block to be loaded in. Called by $904b, $c9ff, $ca4c, $ca98 c469-c478 GET a block with the preparation work being done.....

c479-c4c0 Get a block from disk. Called by $c472 and $cb26 via JMP at $c21a

c4c1-c541 Flips Kernal and I/O in, and sets them up for use. from jump table $c25c

c542-c547 IRQ target, $0314 set by $c4e8 when I/O active

c547 NMI target, $0318 set by $c4f0 when I/O active.

c548-c567 restores I/O registers, location $01, and processor flags to their values prior to being initialized by routine at $c4c1.

c568-c58c two entry points (c56c). First entry point called by $c373, $c5ae, $c49b. Second entry point called by $923a, $9282, $92c2, $c490, $c44b. If $c568, then this routine sets up drive code entry pointer for fast i/o and index to send two bytes to drive. If $c56c, then this routine sets up track and sector, and drive code entry point and index to send four bytes to drive. Pointers in $8b and $8c prior to entry here. Once it enters here, transfers the pointer values to $8863 and $8864. Then 8b/8c points to $8863. $8865 temporary track number storage. $8866 temporary sector number storage.

c58d-c599 called by $c569, $c4a6

c59a-c5a5 check for handshake from drive. called by $c0a6, $c0b2, $c057

c5a6-c5d2

c5d3-c5dd table of values used by the above routine ($c5a6) c5de- beginning of drive code corresponds to $0300 in the drive

c5de = drive $0300 (buffer 0)
c6de = drive $0400 (buffer 1)
c7de = drive $0500 (buffer 2)
c8de = drive $0600 (buffer 3)

entry points:

c5fd($031f)
c6ba($03dc) M-E entry point
c6f0 (0412)
c706 (0428)
c710 (0432)
c751 (0473)
c87a (059c)
c88c (05ae)

fast I/O:

c5fd-c652 drive side sender of fast I/O ($031f in drive RAM).

c653-c69b drive side receiver of fast I/O ($03bd in drive).

c933 end of drive code (drive $0655)

c934-c956 set the current device for the C64 and GEOS operating systems, shut down the FAST I/O in the drive

c957-c96b from jump table $c247 sets the pointers for BAM load, then loads it by JMPing to $c469 via $c1e4

c96c-c98e routine for processing GEOS file type bytes reference by jump table at $c208. $c980 puts GEOS file type byte one in $8416

c98f-c9a6 addresses of routines for various file type processing (for byte position 21 & 22 in the directory entry) c9a7-c9f6 FIND A FILE responsible for constructing the currently active directory entry ($80 bytes at $8400) called from $97eb, $cc65, $c96f, $cc82 This routine seems to handle most of the building of directory entries?

c9f7-ca0f sets a pointer to 18/1, and load first directory block into $8000, then sets pointer in $0c-d to $8002. Called from $909b & $c9b6

ca10-ca59 steps through directy entries in block loaded into $8000-$80ff. If it runs out of entries in this block, it loads in the next block and resets the pointer to the beginning of the new directory block. Called by $c9da, $90da

ca5a-ca7f called by $ca43. Checks to see if a disk is a GEOS disk, and if so, sets up for load of border directory block

ca80-cac6 references VLIR file type flag in the info block $cac7-$caff loads the info block . called by $9120 and $915c, and $cace

cb00-cb92 load data from disk and contruct link table. Called by $9cf8, $915c, $cace via JMP at $c1ff

cb93-cbb1 'Is it a GEOS disk' routine, called by $9014 and $ca68 via $c1de. Uses $848b to flag whether or not it is. ff = yes, 00 = no. Exits with value in .A

cbb2-cbc2 "GEOS Format V1.0" text with a delimiting zero for comparison with disk, excluding the "V1.0."

cbc3-cbd3 open the command channel to the drive

cbd4-cbe7 send characters to current device via $ffd2

cbe8-cbf4 (no reference?)

cbf5-cc20 this is the very first subroutine of GEOS intitilization. Called by $cc3b directly, and this is the only call to it we have ever found.

cc21-ccd1 initialization routine - JMPed from $c003 this is what runs after geos boot

ccd2-ccda sets pointer to data table. called by $cc62, $cc7f, $cc9c

ccdb-cce2 'desktop' text for loading deskTop, delimited by a $00

cce4-ccf7 dialogue box data first byte identifies it as dialogue box ($81) desktop is loaded with these things. $0b second byte (also common) indicates kind. Next two bytes probably positioning commands. Next two bytes point to the text that is to be printed. $81, $0b, $10, $10, $f8, $cc. Text lies at $ccf8. Then $0b, $10, $20, $0e, $cd. $cd0e is where the second part of the text lies. Of the last three numbers (1, 11, and 48), 11 and 48 appear to be position command. The 1 seems to be the OK/Cancel type indicator. 1=OK numbers correspond to sequence at $f44a through $f479.

ccf8-cd30 three text lines used in the dialogue box above. Each delimited by $00

cd31-cd4c Flips everything out, #$ffs $87eb-$87fa, sets pointer to $cf3f, JMPs to $c373. Called by $9168, $cc5f, $cbfa, $ccbe, $f27f

cd4d-cd6a values stored in $d01d (32 byts for VIC chip)

cd6b-cda2 routine that flips in hi-res screen loads $d000 values from $cd4d resets hardware and i/0 vectors exits into joystick input driver comes from $cbf7 and $cc5c cda3-cde0 fills foreground and background with $aa & $55 two screen initializion the screens are at $6000 and $a000

cde1-cdec resets hardware i/o vectors to standard at $0314-$0333 called by $cd99, et.al. $9d66

cded-ce42 initializes color memory at $8c00 sets sprite for sprites 0 and 1 to blue (6), border color to black (0) and pulls the significant sprite data from $bf40 and puts it at $c4c1

ce43-ce4c Shuts down the fast I/O, zeros its flag, puts all but BASIC back in, and JMPs to $fce2. BASIC cold start, the same as a SYS 64738

ce4d-ce72 Zeroes out a target area of RAM. Called by $908b. Referenced by the jump table $c178.

ce51-ce72 Fills a targeted area of RAM with the value stored in $06. $04-5 points to the beginning of the target area. $02-3 contains the number of bytes to write to. Subroutine This routine used to fill the hi-res color RAM at $8c00. Called by routine at $ce13 via table and $fc88 directly.

ce73-ceaa This routine is entered through jump table at $c181, as from the JSR at $0416 in deskTop, or directly, as by the JMP at $cd4a or JSR at $cd85, write values to RAM. Upon entry, $02-3 must be pointing to the data you want written. Byte 1 of your data table is your target address low byte, byte 2 is its high byte. Byte 3 must be the number of characters you which to store. Bytes 4 on are the values you which to send. As an example of such tabels can be seen at $cf3f-$cfa0. the routine uses $04-5 as its indirect storage pointer, $06 as its character counter, and returns with the last address written to in $02-3 ceab-ceba This routine is a workhorse. Called by $9176, $c2dd, $c98b, $cccb, $delc, $de6d, $e330, $e339, $e708, $eff8. It allows the programmer to put the low byte in .A, High byte in .X for storage and indirect JMP ($41).

cebb-cec7 works address in $3d-e. Called via $c2a4 by $dae3, $eef0, $ef23, $fbcb, $fc8e

cec8-ceda writes values into VIC registers, called by $cd92

cedb-cf03 called by $91d6, $c975, $ccc4. Restore pointers for applications

cf04-cf3e save pointers for applications. Called by $c9c6, $91c6, $ccb8

cf3f-cf93 table of initial RAM values

cf3f-cf4d twelve values placed in $2e00-$2e0b, perhaps for photo manager or text manager. Compare their load addresses.

cf43-cf6c twenty eight values placed in $849b-$84b6, a section of GEOS working RAM

cf6d-cf71 two values place in $877d-$877e

cf72-cf75 one value placed in $8808

cf76-cf80 eight values placed in $8ff8-$8fff; sprite pointers

cf83-cf87 two values place in $dc02-dc03 CIA #1, data direction register A and B.

cf88-cf8d three values placed in $dcod-$dc0f: CIA #1 interrupt control register, control registers A and B.

cf8e-cf92 two values place in $dd01-$dd02: CIA #2 data port register B, data direction register A

cf93-cf98 three values placed in $dd0d-$dd0f: CIA #2 interrupt control register and control registers A and B.

cf99-cf9e three values placed in $87d7-$87d9, just prior to Keypress input buffer

cfal-cfff sets PAL/NTSC, initializes system clocks. called by $ccla

d000-dfff i/o when needed, primarily code in RAM underneath
d000-d20f screen graphics data $d210-$d4f7 BSW 9 point default font

d629-d62e three addresses used at $d551-$d55c

d696-d6f5 48 addresses used for JMP ($la)

d99f-da65 table at $d696-$d6f5 references these bit shifting routines

de70-de97 addresses used for JMP ($41)

dfc0-dfee default font used

dfc8-dfee used another font

elbd-e205 create text entry bar sprite. Called by $e075 direct. Referenced by JMP table at $clc0.
e25b-e25f table of values used for subtraction at $e219 and addition at $e22d

e260-e264 table of values used for addition at $e220

e2dc-e35f maskable interrupt request and BRK hardware vector routine e360-e36e main interrupt routine

e456-e45e table of comparison values used at e411

e45f-e467 table of values loaded at $e41b

e468-e46f table of AND values used at $e3ce and $e42c

e470-e477 table of values stored into DATA PORT Register A at $dc00

e478-e4b7 table of values loaded at $e3e4

e4b8-e4f7 table of values loaded at $e3de

efff-f012 two address tables used for indirect JMP ($41).
$efff-$f008 - low bytes loaded at $eff2;
$f009-$f012 - high bytes loaded at $eff5

f189-f1ba table of 25 screen line addresses:
$f189-$f1a1 - low bytes loaded by $f12f, $f146, $f160
$f1a2-$f1ba - high bytes loaded by $f136, $f14f, $f167

f1bb-f23e dialog box (see $c256) called by $fce4, $cc90, $08ac, $0a0c, $0d60, $0de8, $0e22, et.al. from desktop

f23f-f264 table of 19 addresses for 19 different dialog boxes types. Numbers $07-$0a are ringers, probably for future additions.
$f23f-$f251 low bytes loaded at $f20a
$f252-$f264 high bytes loaded at $f20d

f329-f32e six byte table pointed to by $f2f2-$f2f9

f32f-f346 (see $c2bf)

f3af-f3d1 eleven three byte data segments used by the routines at $f35d-$f371 and $f372-$f389 to save and restore memory. Byte one and two are low byte/high byte of an address placed at $06-$07 and byte three is the number of characters to be moved.

f60c-f638 value table used by subroutine at $f420

f420-f449 routine to call dialogue box buttons.

f44a-f478 addresses and information used to call graphics for dialogue boxes. sequence = 1=ok, 2=cancel, 3=yes,4=no, 5=open, 6=disk

f47a-f497 dialogue box index routine, stores box number at $851d.

f7dd-f830 "NO" dialog box graphics data
f831-f885 "YES" dialog box graphics data
f886-f8db "OPEN" dialog box graphics data
f8dc-f92f "DISK" dialog box graphics data

f9cd-f9d8 value table loaded at $f9b8

fa54-fa6c values for SID registers

fa6d-fa77 math or bit shifting routine. .X=zero page address to be shifted, .Y=number of shifts to perform. Shift left.

fa78-fa82 math or bit shifting routine. .X=zero page address to be shifted, .Y=number of shifts to perform. Shift right.

fb82,
fb84-fb9f generic self modifying memory mover

fc44-
fc46-fc67 generic self modifying memory comparer

fcbd- fd07 BRK routine. Called by $e2e6 which is an indirect JMP off the BRK vector at $84af. This routine probably prints "System error near ..."

fd08-fd27 dialogue box information for "System Error near xxxx", pointed to by $fcde, $fce3

fd74-fd7f text: "GEOS KERNAL" with zero delimiter.

fd80-fd9e table of values in one seven byte chunk and three eight byte chunks.

fd80-fd86 referenced by $ee8f-$eec2

fd87-fd8e referenced by $970d, $9e34, $9e44, $9e51, $9e69

fd8f-fd96 referenced by $d582, $d5e2, $d718, $ecc5

fd97-fd9e referenced by $d591, $ecbb

fd9f-fe7f looks suspiciously like the remnants of a previous assembly of the GEOS source code. No apperent references are made to the code in this range, and some coming out make no sense, like the little routine at $fdf5-ff

fe80-fff9 Input driver (joystick file) $fe89-$fe90 joystick working RAM

fffa-ffff hardware vectors

fffa-b NMI point to $c2e4

fffc-d RESET $c2e4 (initially off the disk, the reset may be pointing to $cc21,

fffe-f IRQ $e2dc

e000-ffff Standard Kernal as needed

NOTES

Appendices

APPENDIXES

ERROS & PROBLEMS

In normal, careful use, errors should be rare when working with GEOS. Most errors that do occur usually involve the disk drive, making backups of data disks even more important.

GEOS is a new operating system, and very disk intensive. The current version is only the second release, in contrast to the number of releases seen by CP/M, MS DOS, and other disk based operating systems. The best approach is to prevent the problems from occuring.

GEOS can, and will, get confused if you switch disks while in an application. If you are working within geoWrite and load "document" from the disk "letters," take out the "letters" disk and replace it with "term papers," odds are good that GEOS will crash and you might corrupt one or both disks. This is especially true if you do not give each of your disks different names. Rename and keep track of your disks.

Three kinds of errors can occur while working with GEOS. The first, and simplest, causes GEOS to inform you that there is an error and prompt you with an [OK] box. In this case, no damage is done: select [OK], exit, and correct the problem.

The second kind of error is insidious, but may be harmless if caught quickly enough. These errors occur when you do too much with GEOS, or try to do something for which GEOS is not fully prepared. The normal indication of this kind of error is garbage appearing somewhere on the screen, system lockup, or endless loop. Even if the garbage goes away, GEOS is confused. Exit the application, or deskTop, immediately, reboot, and select Disk/validate. With the exception of trying to erase a locked file from the border of deskTop, the disk data should all be intact and you can procede without too much fear.

The third kind of error is fatal to the current session, and damaging. GEOS will open a dialogue box informing you of an error "Near xxxx." In this case, you will have to turn off your computer and reboot GEOS. If you were working within an application, the document you were working on will most likely be corrupted.

If this should happen, the easiest thing to do is reformat both disks and start over. If either of the disks have enough important data on them, copy files - not the whole disk - to fresh disks. Then check each of the files to make sure that they are intact. Any corruption of the data on a disk is likely to be nearly invisible and may not become apparent until several work sessions later.

One of the problems that most users will encounter is the "Printer File not on disk" error. This is a simple, but bothersome, quirk of the operating system. When GEOS first loads, it searches the boot disk for the first input driver and the first printer driver and uses these as the defaults. If your printer does not use the first driver as found on the boot disk, and your work disk does not have the default printer driver on it, you will get this error.

The safest solution is to select the GEOS/select printer command the first time you insert your work disk each session. This can be rather inconvenient. If you do not wish to create another boot disk with the utility in section two, then you may alter the original boot disk. FIRST BACKUP THE ORIGINAL. Then rearrange the order of the printer driver files by moving icons to the border and back. You need only move the first printer driver file and your desired printer driver file to the border, then move your desired printer driver file back onto the deskTop notepad page.

PRINTERS

In order to offer the most flexibility and take full advantage of most printers, GEOS works on the premise that you will use an 80 dot per inch (dpi) printer. At 80 dpi, geoWrite will automatically create 1.2 inch left and right margins and geoPaint will create 8 inch wide documents. If your printer provides less than 80 dpi resolution, the geoWrite margins will be proportionally smaller, and part of the geoPaint document data will not be printed. For more information on this, please refer to page 1-106.

Following is a table of printers and the appropriate printer driver. For all Centronics type printers, the table assumes that your interface is set to transparent mode. This is the mode that allows all data sent from the computer to be passed on to the printer without modification. If you are using a Centronics type printer, you may also attempt to use the 1525 emulate mode provided by the interface and use the MPS 801/803 or MPS 1000 printer drivers. This may cause some loss in print quality or resolution.

INPUT DRIVERS

GEOS was originally designed to provide the look and feel of the Amiga's Workbench and the similar user interfaces on the Apple MacIntosh, the Atari ST, and GEM or Windows on MS-DOS machines. Each of these systems either has, or has available, an analog mouse. The Commodore 64 is not so lucky.

In contrast to digital joysticks and digital mice which provide a narrow range of signals - usually one byte digital values in the range of 0 to 7 - analog devices provide a much wider range of values that more closely approximates some reality. A typical example of an analog device is the standard clock, because the hour hand of the clock is similar, analogous, to the way the sun goes across the horizon.

A analog device coupled with a digital computer needs an analog to digital conversion. The frequency that the analog signal is measured in large part determines the quality of the effect. High quality digital sound samplers, for example, "take a picture" of the sound several thousand times each second in order to provide a high quality reproduction. The digital sound samplers that are available for the Commodore 64 only sample the sound from a few hundred to a few thousand times each second, and the reproductions sound like an old-time radio.

In any case, an analog input device is not yet commercially available for the Commodore 64. Such devices have been designed, and software drivers for such devices have been written, but it is not certain that they will ever become commercially available. Standard digital devices such as joysticks are still selling in the ten dollar to twenty dollar range, digital mice are selling for thirty dollars to fifty dollars, and analog mice would probably sell for seventy dollars to one hundred dollars. Such a price makes it unlikely that analog mice will ever be mass produces.

In the meantime, Berkeley Softworks and third parties are developing and releasing drivers for other types of input devices. Berkeley has already written and released a driver that allows the use of the Inkwell Flexidraw (tm) Lightpen, and a driver that can be used with the Koala Technologies KoalaPad (tm) or Suncom Communications' Animation Station (tm) sketchpads. These drivers for alternative devices are patches to the original GEOS code, and so do not offer full control. A third party has introduce a sketchpad driver via Quantum Link, and more such drivers are likely to be forthcoming.

The best, and most consistent, driver remains the digital joystick or mouse driver. Although slow and cumbersome at times, it provides the full range of motion and control within GEOS, is fully supported, and widely available at a reasonable price.

DISK SECTOR EDITING

The disk drive is certainly the most popular peripheral for Commodore computers. Most users are satisfied to use disk drives without a full understanding of the way it stores information on the disk. This is no handicap until something happens to an important file.

Sector editors allow you to look indirectly at the information stored on disk. A locally popular sector editor, various incarnations of Disk Doctor as modified by Kevin Hisel, provides a character graphic representation of the information of the disk, one block at a time. Other sector editors represent the data in hexidecimal or mnemonics. Most sector editors allow you to alter the data on the disk.

For more information on the way information is stored on disk, study the various disk maps in section 3.

As with so much else involving computers, you can easily become frightened at the prospect of something new. Sometimes this is a reasonalble fear - digging inside a video monitor can be deadly. Sector editing, however, is harmless. The worst damage you can do is ruin the information on the disk. This is easily prevented or cured by practicing on an unimportant disk, or a duplicate of an important one.

Many hours of work can often be easily recovered from a corrupt disk if you are familiar with sector editors. Check with your local user's group, bulletin boards, or on-line services for a sector editor, and play around!

TELECOMMUNICATIONS

With the price of computers and peripherals - especially modems - telecommunicating is becoming increasingly popular. Commodore users are participating fully, in spite of attempts by Commodore, Berkeley, and Quantum Link to make things hard. All three of companies seem intent on making closed systems.

Commodore does not offer a true RS-232C serial communications port on its 8-bit machines - you must either buy a Commodore compatible modem or an interface to use a standard RS-232C modem. This, however, is a mixed blessing as quality 300 baud modems have been available for as little as twenty dollars, and 1200 baud modems for as little as one hundred and twenty-five dollars, much less than the price for standard modems.

Berkeley has created a new file structure that does not lend itself to the transmission of files over telephone lines. A GEOS VLIR file must first be converted into a sequentially linked file before it can be transmitted, and must then be converted back to VLIR format before it can be used. Berkeley has, however, addressed this by providing "CONVERT" on Quantum Link to make the transmission and reception of GEOS document files possible.

Quantum Link, however, is a totally closed system. Users MUST have a Commodore 64, 64C, or 128 PC and Quantum Link's proprietary software to log on to Quantum Link. The Q-Link software does not provide all the features that users might want and does not lend itself to modification. Q-Link is not accessible to Amiga, VIC, or PET users.

In spite of all this, telecommunications is a growing area. On-line services such as Compuserve and Delphi provide an open system for users of all computers. Bulletin boards are usually more limited, but many communities have several BBSs that provide for a range of interests.

Bulletin boards are an excellent resource for information, programs, and comraderie. If you are interested, check with your local computer shop or users group.

Telecommunicating is generally fairly easy. Almost all communciations services - major commercial services and local bulletin boards - follow the de facto standard for basic communications protocol. A communications protocol simply defines what sequence, and how many, ones and zeros make up each character. This standard specifies that data will be encoded with 8 data bits, 1 stop bit, no parity, and at full duplex. Don't worry about what all this means unless you have trouble connecting.

Once you get two computers communicating, you can exchange messages with other users of the service and get a wide variety of quality programs virtually free.

The exchange of files, whether data or program, requires that a secondary communications protocol be followed. Files are usually exchanged almost automatically. The computer that has the file says "Here comes piece #1!" and the computer getting the file says "I'm ready!" The first computer sends piece #1 and says, "That was piece #1!" and the second computer says "I got it!" and so on, and so on, until the whole file is sent.

To make this work, both files must be using the same "file transfer protocol." The file transfer protocol simply defines what computer characters will be used to say "Here it comes!" "I got it," etc.

Most computers that allow telecommunication at all support the Xmodem (Christianson) protocol. Xmodem has been around for several years and has been highly refined. Unfortunately, there are two variations of Xmodem popular enough to cause some confusion - Checksum and CRC. Open commercial services usually support both automatically, but many smaller services and local bulletin boards only support one of these, and don't specify which. The SYSOP (SYStem OPerator) may or may not be able to help you.

Some programs for Commodore computers support other file transfer protocols as well. The most popular protocol for use between Commodore computers is the Punter protocol. Again, this has two versions, with the older simply designated "Punter" and the newer designated "Punter C1." Most Commodore bulletin boards support one of the two Punter file transfer protocols.

Kermit is a relatively new file transfer protocol that is not yet in widespread use, but is growing in popularity.

MODEMS

As noted above, the Commodore VIC series (VIC 20, C64, and C128) does not follow the true RS-232C standard. In order to connect an RS-232C modem to a VIC series computer, you must purchase an interface such as the Omnitronix Deluxe RS232 Interface for Commodore Machines.

Several good modems have been made specifically for the Commodore VIC series of computers. Commodore has made the VICModem (1600), 1650, 1660, and 1670 modems. Other modems have been made by third parties.

Modems are picky about how you talk to them. The Commodore User Port (on the back left as you face the front of the keyboard) has twenty-four pins, and not all modems use the same pins for the same thing. This will never damage your modem or your computer - unless you try to put the modem in upside down - but it does mean that not all modems will work with all software. Modems are also touchy about timing.

Two basic standards have evolved for Commodore compatible modems. For 300 baud communications, most software supports the Commodore 1650, while for 1200 baud communications, most software supports the 1670 and other Hayes type modems. Very few commercial software packages will not support these two modem types.

Non-standard modems, such as the Commodore 1660, Mighty-Mo, Hes Modems, and Anchor modems, usually come with some kind of telecommunications software; it is not necessarily very good. Commercial telecommunications programs may or may not support these modems, and public domain software for these modems is sparse. Unless there is some compelling reason to buy one of these, it is best to stick with 1650 compatibles, or 1670 and Hayes compatibles.

BIOGRAPHIES

Stephen R. Gast - Steve started with computers relatively early, learning to program in Fortran IV on PDPs in 1970. He purchased his first Commodore 64 in December of 1982 and founded the Champaign-Urbana Commodore Users Group in October 1983. He is a pragmatic programmer, and proficient in enough computer languages to choose a language suited to the task at hand. Over the past three years, he has merged the DOS Wedge with Tiny Aid, added several refinements to author Super-Aid, and then worked with Richard Rollins to produce one of the finest programmers' utility packages available, Super-Aide 3.1. He has attended the Lincoln College Computer Camp every year. Stephen graduated from the University of Illinois with a degree in Biology, with minors in Mathematics and Physics, and enjoys insect collecting, wood working, string art, protozoology, and science fiction. He supports his wife, Letha, two children, Charles and Martha, and hobbies with his own pet store in Champaign.

Kevin Hopkins - Kevin bought a Commodore 64 in June of 1984, joined CUCUG four months later, and has been digging into code ever since. Although few programs bear his name, several have been enhanced by his support. In addition to collaborating with Richard and Steve on Super-Aide 3.1, he has made several scrolling versions of a modified Xmon and combined the proofreaders from Compute!'s Gazette and Ahoy into a singly utility. Kevin was formerly active as a journalist and environmentalist, and as lead guitar player for local bands has opened several concerts for groups such as Duran Duran, Stray Cats, U2, Berlin, The Tubes, and others. A 1976 graduate of the University of Illinois in Liberal Arts and Sciences, Kevin is currently a poet, a stage carpenter at the Krannert Center for the performing arts, and computer hacker extraordinaire.

Richard Rollins - Richard's interest in computers started with his purchase of the VIC 20. After prematurely completing a community college course in BASIC, he met Mr. Gast. Since then, he has attended the Lincoln College Computer Camp, written several BASIC programs for local users and schools, and co-authored Super-Aide 3.1. He is currently working to master number crunching on the C64 and starting to explore 68000 machine language. Richard has been a member of CUCUG for two and one-half years, which he currently serves as Novice User Liaison. Richard considers himself a hacker in the truest sense - hacking at code to get the most out of the machine. He enjoys making tools to help other users get the most out of the machine. To support his wife, Vicky, three daughters, Amy, Beth, and Rachel, and his computer hobby, Richard works as an ASE Certified Auto Mechanic.

Tim Sickbert - Tim is a relative newcomer to the Commodore world. Anticipating the resumes he would need to send after graduation from the University of Illinois, he asked for a Commodore 64, 1541, and daisy wheel printer as a graduation gift. Within a month, he volunteered to take on the Champaign-Urbana Commodore Users Group (CUCUG) newsletter and two months later started his first real job as Editor of the Midnite Software Gazette. An accomplished user, Tim has become proficient with a large number of word processors, telecommunications programs, and utilities. With the help of Jim Oldfield, Steve Gast, and others, he has learned enough about the internal workings of Commodore machines to make sense of it. Tim is newly married, and enjoys river canoeing, snow skiing, going everywhere barefoot, and science fiction.

James A. Oldfield, Jr. - Jim started with Commodore computers back in 1978. With Jim Strasma, he started the Central Illinois PET Users Group (CIPUG) newsletter, which grew into the Midnite Software Gazette, the oldest independent magazine in North America for users of Commodore brand computers. In addition to his experience in programming and publishing, Jim has become active in telecommunications and computer market analysis. Debbie, his wife of fourteen years, occasionally works part time for the Midnite to help support their family of three boys.

ACKNOWLEDGEMENTS

This book is the product of a team. A small core of people are responsible for the majority of the book, with help and contributions from several others.

Tim Sickbert is responsible for most of the text as it appears and for coordinating the research and development.

Steve Gast and Richard Rollins are responsible for making this book complete. These two are responsible for the text and graphics translation utilities, and for decoding both the file and data structure. Richard also wrote the BUILD, UNBUILD, UNSCRATCH, and EASY LOAD utilities. Both Steve and Richard have performed far beyond any reasonable expectation.

Kevin Hopkins has worked almost alone to provide all the information contained in the memory map. He has done an excellent job of giving the core of the GEOS operating system. All programmers will appreciate his tenacity and the quality of the work he has done.

Jim Oldfield provided, and continues to provide, oversight and vision to the Midnite Press/Midnite Software Gazette. He is materially responsible for the telecommunication program included, and for providing the opportunity for the rest of the team to get together and work on this book.

Dr. Richard Immers provided information crucial to understanding the GEOS BOOT file, and has been a resource to all disk related projects, either personally or through his excellent reference book, "Inside Commodore DOS."

Wayne Schmidt provided information regarding the source files supported by the graphics conversion utility, and provided all the graphics we used to test it. His work can be seen regularly just about anywhere you find Commodore 64 graphics.

BIBLIOGRAPHY

Immers, Dr. Richard. Inside Commodore DOS

GEOS User's Guide V1.2. Berkeley Softworks, Berkeley California. 1986

Hildon, Karl J. H. The Complete Commodore Inner Space Anthology. Milton, Ontario, Canada. Transactor Publishing. 1985

Commodore 64 Programmer's Reference Guide. Commodore Business Machines and Howard W. Sams, 1982.

Angerhousen, Michael, Dr. Achim Becker, Lothar Englisch, and Klaus Gerits. The Anatomy of the Commodore 64. Grand Rapids, Michigan. Abacus Software, 1983.

Every effort has been made to insure complete and accurate information concerning the material presented in this book. However, The Midnite Press/Midnite Software Gazette can neither guarantee nor be held legally responsible for any mistakes in printing or faulty instructions contained in this book.

All the information herein contained regarding GEOS (tm) pertains to GEOS V1.2 as marketed and sold during the fall of 1986. Subsequent revisions and versions of GEOS are likely to differ in some respects.

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL
First Class Permit 1145 Champaign, Illinois

POSTAGE WILL BE PAID BY ADDRESSEE

Midnite Press
1212 Hagan
Champaign, IL 61820

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL
First Class Permit 1145 Champaign, Illinois

POSTAGE WILL BE PAID BY ADDRESSEE

Midnite Press
1212 Hagan
Champaign, IL 61820

Midnite Press · Midnite Software Gazette

Please send me

□ Complete Programs and Utilities Disk for
HOW TO GET THE MOST OUT OF GEOS **$9.95**

□ 12-issue subscription to the
Midnite Software Gazette **$23.00**

For FAST service, call
(800) 222 - 4441
to order

Name ____________________

Address ____________________

City ____________________ ST ________ Zip ____________

Phone ____________________

□ Payment Enclosed □ COD
□ Visa □ Mastercard

Card number ____________________

Signature ____________________ Exp Date ___/___

Check and bankcard orders please include $2.50 shipping and handling.
C.O.D. orders add $2.50 shipping and handling, plus $2.00 for C.O.D. charges.
Illinois residents please add 6% sales tax.

Allow 4 to 6 weeks for delivery.

Dealer Inquiries Welcome

Midnite Press · Midnite Software Gazette

Please send me

□ Complete Programs and Utilities Disk for
HOW TO GET THE MOST OUT OF GEOS **$9.95**

□ 12-issue subscription to the
Midnite Software Gazette **$23.00**

For FAST service, call
(800) 222 - 4441
to order

Name ____________________

Address ____________________

City ____________________ ST ________ Zip ____________

Phone ____________________

□ Payment Enclosed □ COD
□ Visa □ Mastercard

Card number ____________________

Signature ____________________ Exp Date ___/___

Check and bankcard orders please include $2.50 shipping and handling.
C.O.D. orders add $2.50 shipping and handling, plus $2.00 for C.O.D. charges.
Illinois residents please add 6% sales tax.

Allow 4 to 6 weeks for delivery.

Dealer Inquiries Welcome

HOW TO GET THE MOST OUT OF GEOS . . .

— is for the new user who wants to know more about GEOS and how to make it work for him;

— is for the advanced user who wants more flexibility than GEOS alone offers;

— is for the programmer who wants to write his own utilities to extend his control over GEOS;

— is the book you need when the manual is not enough.

HOW TO GET THE MOST OUT OF GEOS gives you the information that you need to use this powerful program to your best advantage. A team of experienced users and programmers examined GEOS inside and out to give you:

— shortcuts to make your work go faster and smoother;

— undocumented features that you will want to take advantage of, or avoid;

— translation utilities to import text and graphics from other popular programs;

— conversion utilities to let you transmit GEOS files to your friends via modem;

— the GEOS file structure, so programmers can write utilities for their own special needs; and

— a map of important memory locations so advanced programmers can create their own custom utilities.

From the Editors of the Midnite Software Gazette, the first independent U.S. magazine for users of Commodore brand Computers.

Midnite Press·Midnite Software Gazette
1212 Hagan, Champaign, Illinois 61820

ISBN
#0-941377-00-8