Using ProDOS Part 1

The Multilayered DOS

by Lee Swoboda

Last year Apple unveiled ProDOS (Professional Disk Operating System), the latest, greatest disk operating system for the venerable Apple II. Although similar to DOS 3.3 on the outside, and upwardly compatible with it, ProDOS represents a major redesign on the inside.

ProDOS is Apple's solution to the deficiencies of DOS 3.3. Among its distinct advantages, ProDOS is up to eight times faster than DOS, can be used with hard disks, and is compatible with SOS, the Apple /// operating system.

Those familiar with events in the Apple world may recognize similarities between the introductions of DOS 3.3 and ProDOS. But ProDOS is a much more radical departure from its predecessor than DOS 3.3 was from 3.2. ProDOS has replaced DOS 3.3 as the standard Apple II operating system and is included with all new Apple disk drives.

The first in our six-part series on Apple's latest DOS focuses on its hierarchical file structure.

In a six-part series, I'll explain the wonders of ProDOS. The topics will include:

1) The ProDOS File Structure
2) Using ProDOS from the Built-in Menus
3) Improved ProDOS Commands
4) New ProDOS Commands
5) Using ProDOS from BASIC
6) Compatibility of ProDOS with DOS 3.3

The series is written for the novice who uses ProDOS as a first disk operating system, as well as for the experienced Apple II computer user who wants to upgrade DOS 3.3. It will stress hands-on instruction to understand how each ProDOS feature works. But first, whether you're a neophyte or an old hand, you'll need some background information.

Getting Down to Basics

ProDOS is a scaled-down version of the Apple /// Sophisticated Operating System (SOS). The SOS and ProDOS floppy disk formats are nearly identical, even to the point of generating exchangeable textfiles. (You cannot, however, swap BASIC programs, because Apple II's Applesoft and Apple ///'s Business BASIC are incompatible.)

When you buy a new Apple disk drive, you automatically get the ProDOS User's Kit, which includes the

ProDOS User's Manual, the *ProDOS Supplement to the Apple //e Owner's Manual*, and the ProDOS User's Disk.

When you buy a //c, you get ProDOS on the System Utilities disk, and a System Utilities manual. If you already own a II Plus or //e, you can purchase the ProDOS User's Kit (Apple product A2D2010) separately for $40. Unlike the DOS 3.2 to DOS 3.3 conversion, which required changing two integrated circuits on the disk controller card, adding ProDOS to your Apple requires no hardware modifications.

ProDOS requires an Apple //e, a //c, or a 64K Apple II Plus (48K II Plus with a 16K "language" card). According to the *ProDOS Technical Reference Manual*, you may use ProDOS in a 48K Apple II Plus, but Apple doesn't recommend it. If you have a regular Apple II (with integer BASIC in ROM), you can't use ProDOS *even though you have a language card.*

Because ProDOS is a *disk* operating system, you must, of course, have at least one Apple-compatible disk drive. (Two are better.) Be cautious about compatibility, however. Apple's Disk II is a 35-track drive. If you have a "super" drive, which has more tracks, you may be unable to use ProDOS. Ask your dealer if your disk drive can handle ProDOS. If there's any doubt, test ProDOS on your system before you buy, and check the dealer's return policy.

Tree-Structured Files

The most dramatic difference between DOS 3.3 and ProDOS is the latter's tree-structured file system. **Figure 1** illustrates the typical DOS 3.3 file arrangement. Each file has equal status. To use a botanical analogy, all files "grow" like blades of grass out of the disk at ground level. When you catalog a disk, the computer simply lists the files in the order you created them (unless you deleted, then added, some).

Table 1. *Rules for defining a ProDOS pathname.*

up to 64 characters
(including slashes)

/volume.directory/subdirectory/file.name

file.name — Default: none
Range: see below

subdirectory — Default: prefix subdirectory
Range: see below

volume.directory — Default: prefix volume directory
Range: Up to 15 characters (excluding slashes)
A–Z
a–z (ProDOS automatically converts to uppercase)
0–9
period
first character must be a letter

ProDOS, however, organizes your files in a hierarchy. To continue our botanical analogy, imagine all the files on a ProDOS volume (floppy or hard disk) as apples on a tree. To pick an apple, you climb up the trunk and then inch out along progressively smaller branches until you reach the fruit. ProDOS follows this concept to organize files. **Figure 2** represents the simplest type of ProDOS directory, and **Figure 3** a more complex one.

The trunk of the tree is the *volume directory*. Just as branches radiate from the trunk, subdirectories and filenames on a disk volume "grow" from the volume directory. To reach a ProDOS file, you must follow a path "up" the volume directory and "along" any subdirectories to the file. This route is called the *pathname*, and consists of a volume directory name, any subdirectory names, and the filename. The entire pathname is analogous to DOS 3.3's filename. To define a file in ProDOS, you need to specify the entire pathname.

For example, you might assign the name PETS to a ProDOS disk that will store information about household animals. Under the categories DOGS and CATS, you could then store the respective information about those species. **Figure 4** shows how your disk would be arranged.

But suppose you wanted to store more detailed information, not just about dogs and cats, but about specific

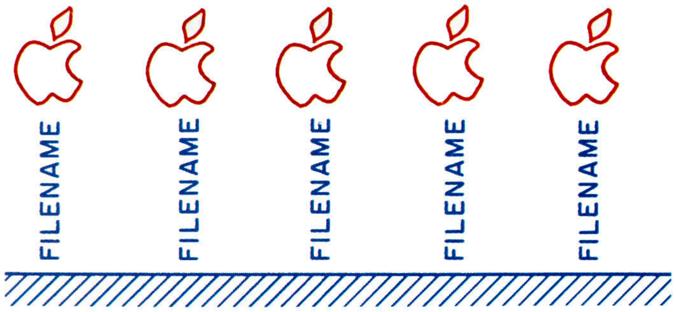


Figure 1. *DOS 3.3 directory.*

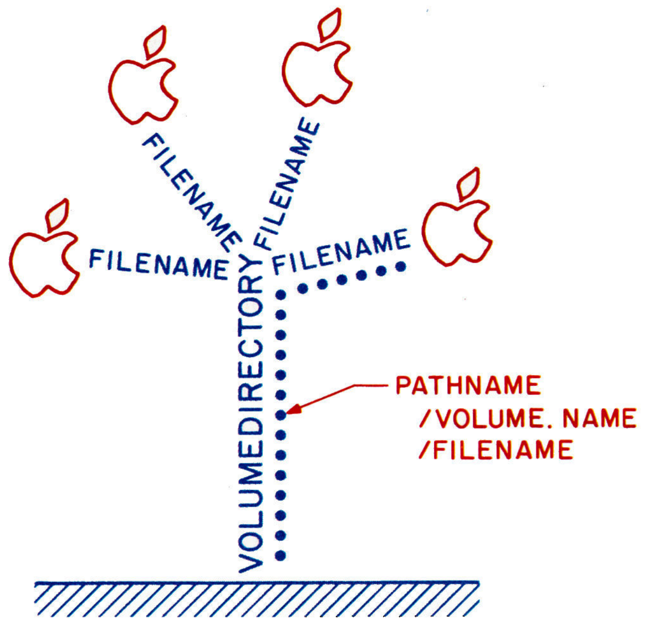


Figure 2. *The hierarchical structure of a ProDOS directory demonstrates how a volume is organized.*

breeds? You could set up subdirectories named DOG and CAT, rather than just files by those names. Then, you could name files in those subdirectories for different breeds, as shown in **Figure 5**.

If you're *really* serious about pets, you might want to keep information about different varieties within each breed. One way to do this would be to name files that correspond to both the breeds and varieties (**Figure 6**). You could also set up a *second level* subdirectory for breeds, then name the files for the varieties (**Figure 7**).

ProDOS generously provides both these options. If you were dependent upon DOS 3.3 for this exercise, you'd have to lump dogs and cats together (a possibly precarious situation) into one group of files on the disk. (See **Figure 8**.) You could distinguish among dogs, cats, breeds, and varieties only by the filename, but compare **Figure 8** to **Figure 7**. See how much more logical ProDOS's tree-structured file arrangement is?

The Hard Disk Advantage

DOS 3.3's one-level file arrangement is adequate for the relatively small storage capacity of floppy disks. But ProDOS, with its tree-structured file arrangement, can handle the thousands of files stored on hard disks. When you tell ProDOS to set up a subdirectory, it reserves part of the hard disk large enough to store the files in that subdirectory. In effect, ProDOS turns a hard disk into a group of variable-size floppy disks, each of which can be located by naming its subdirectory.

Pathnames

As I mentioned, to define a file in ProDOS, you must name its volume directory, subdirectory(ies), and filename. This sequence of names is called the *pathname*. Just as DOS 3.3 has rules for naming files, ProDOS has rules for defining pathnames. Refer to **Table 1** as you read these rules:

- The pathname must contain all the elements that define the file's location: volume name, subdirectory name(s), and filename.
- These elements must be set off by slashes (/).
- The elements can't contain more than 15 characters each (excluding slashes).
- The first character of each element must be a letter.
- Remaining characters can be letters, numbers, or periods.
- You can use lowercase characters, but ProDOS will convert them to uppercase.

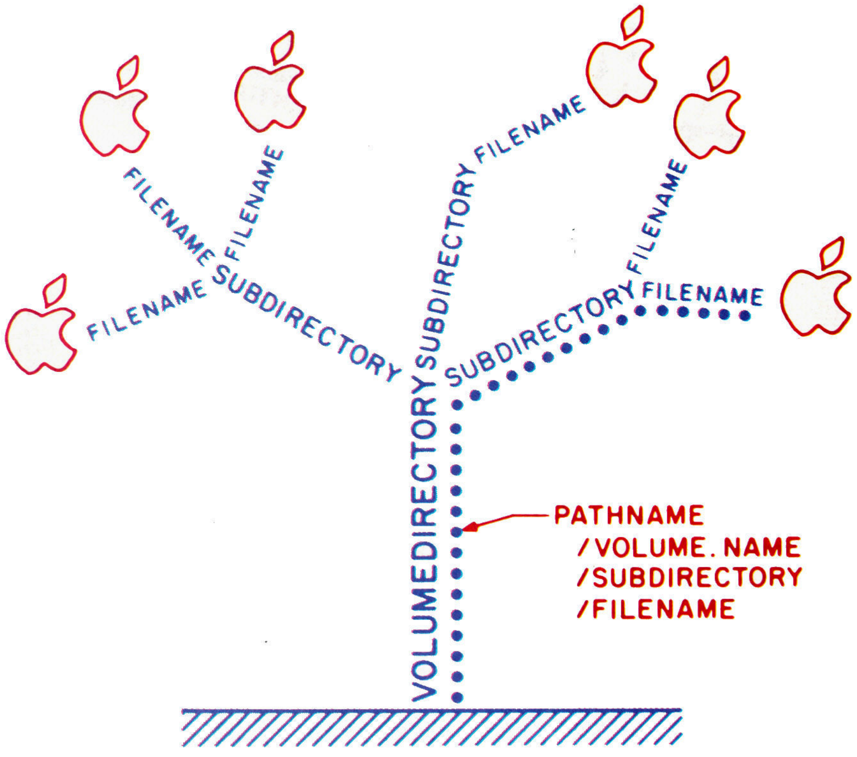


Figure 3. *ProDOS directory represented as a tree.*

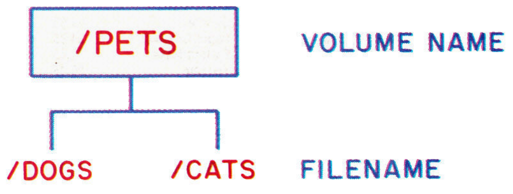


Figure 4. *One-level ProDOS directory.*

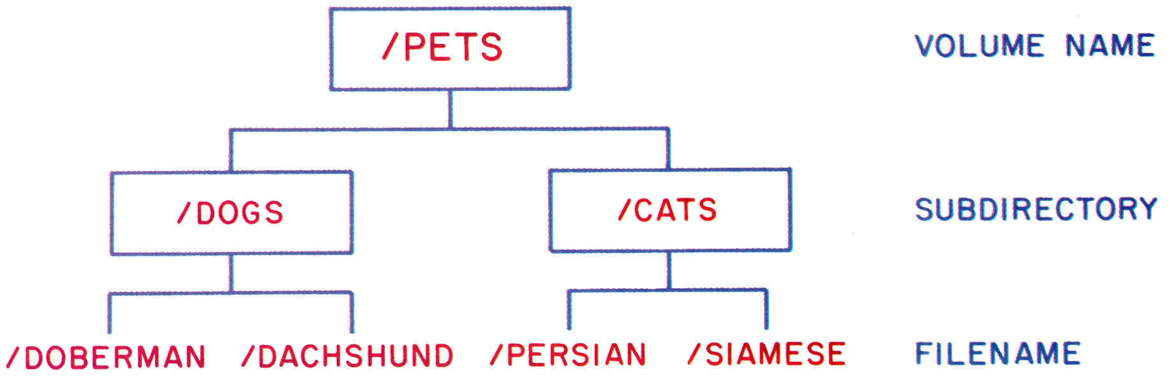


Figure 5. *Two-level ProDOS directory.*

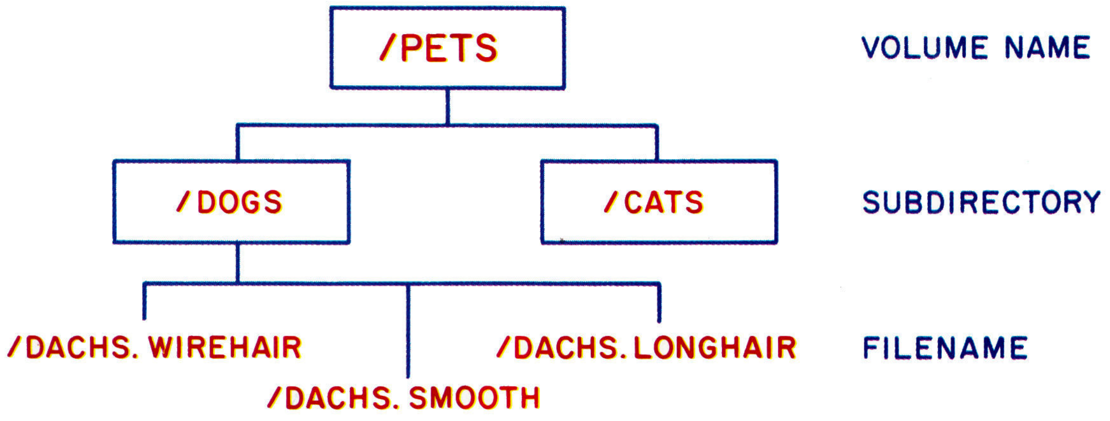


Figure 6. *Two-level ProDOS directory with expanded filenames.*

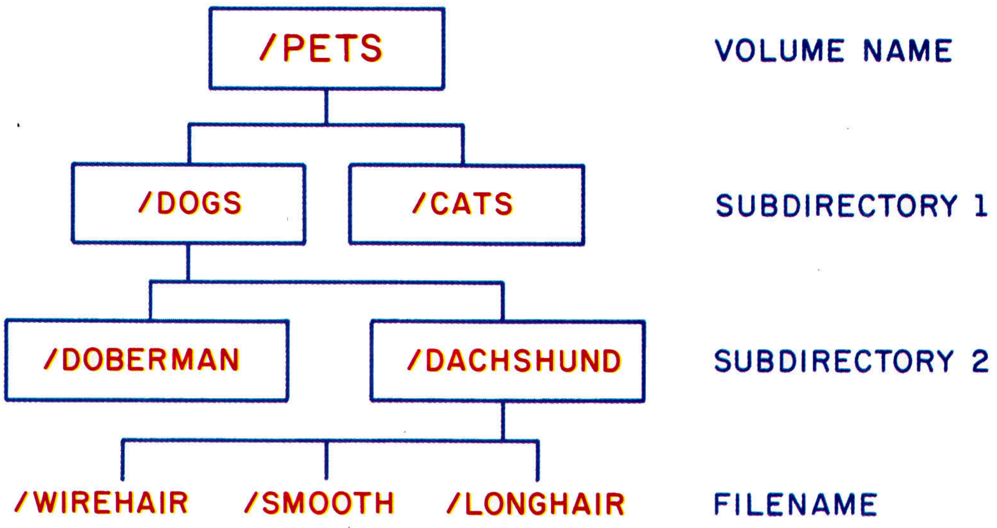


Figure 7. *Three-level ProDOS directory.*

Table 2. *Valid and invalid pathname examples.*

Valid Pathnames	Invalid Pathnames	Violated ProDOS Rule
/USERS.DISK	/APPLE'S.PRODOS	Only punctuation allowed is a period
/MY.OWN.DISK	/APPLE COMPUTER	Spaces not allowed
/LETTER.7.13.84	/3.BLIND.MICE	First character must be a letter
	/APPLE.MACINTOSH .COMPUTER	Fifteen characters maximum

Table 3. *ProDOS and DOS 3.3 filenames comparison.*

ProDOS Filenames	DOS Filenames
Up to 15 characters	Up to 30 characters
Begin with a letter	Begin with a letter
Numbers, letters, periods only	Any character (including control characters)
No spaces allowed	Spaces allowed

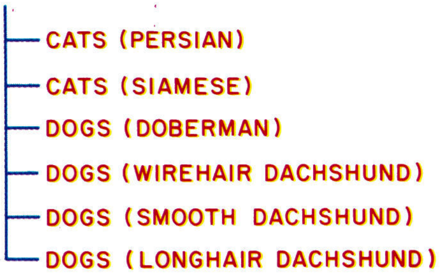


Figure 8. *DOS 3.3 directory.*

● The pathname can't contain more than 64 characters (including slashes).

Refer to **Table 2** for examples of valid and invalid pathnames. **Table 3** summarizes the differences between DOS 3.3 and ProDOS filenames.

If you had to define the entire pathname every time you accessed a file, you would soon tire of typing. To make it easier to use certain files repeatedly, ProDOS allows you to instruct the computer to temporarily remember a part of the pathname called a *prefix*. The computer attaches this prefix to all files you specify until you give it another prefix to remember. For example, if you instructed ProDOS to attach the prefix /PETS/DOGS to a filename, you could access the dog information in **Figure 7** by typing in the breed and variety of dog. If you typed /DACHSHUND/WIREHAIR, ProDOS would attach the prefix to yield the filename /PETS/DOGS/DACHSHUND/WIREHAIR. Or, if you set the prefix to PETS/DOGS/DACHSHUND, you could access the same file with the filename /WIREHAIR.

Well, I've "put on the dog" enough for now. Understanding its file structure is the most important step toward understanding ProDOS. Next month, in Part 2, I'll discuss the techniques for using ProDOS from built-in menus. Join me then.■

Address correspondence to Lee Swoboda c/o Padapple Computer Consulting, 1451 NE Paulson Road, Poulsbo, WA 98370. Representative questions will be answered in future articles in this series or in inCider*'s Letters department.*

Using ProDOS Part 2

Menu Magic

by Lee Swoboda

Harness the power of Apple's new operating system through its menu-driven utility programs.

"Using ProDOS" is a six-part hands-on tutorial describing the use of ProDOS, Apple's newest disk operating system for the II family. In February, we examined the file structure of ProDOS. This month we'll start using ProDOS utilities (special-purpose programs) from the built-in menus.

ProDOS comes in two flavors. You get Version 1.0.1 when you buy a Disk II or a DuoDisk drive. It is found on the ProDOS User's Disk that comes bundled with the hardware. Version 1.0.2 is sold with Apple //c's. It comes on the disk called System Utilities. The differences between ProDOS 1.0.1 and 1.0.2 lie in the way you access the ProDOS utility programs. For the most part, however, they do not differ in the operations they perform.

In the discussion that follows, Version 1.0.1 will be synonymous with the User's Disk, Version 1.0.2 will be synonymous with the System Utilities disk, and the master disk will refer to both.

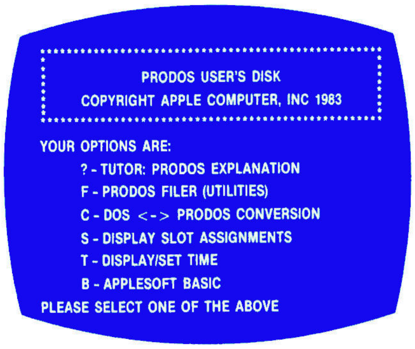
You boot ProDOS using the same procedure used to boot DOS 3.3. If your computer is off, put the master disk into drive 1 (the internal drive on the //c) and turn the computer on. If it already is on, place the disk in drive 1, type PR#6, and hit the return key. Depending on the version of ProDOS you have, the computer will display either the menu shown in **Figure 1** or that shown in **Figure 2**.

The ProDOS Main Menu

DOS 3.3 starts you off with a "hello" program; so does ProDOS. However, unlike DOS 3.3, which allows you to name the hello program, the initial ProDOS program is always called Startup. With your master disk, Startup loads the utility programs and then displays the main menu. This menu enables you to select ProDOS utility functions without having to know the ProDOS commands.

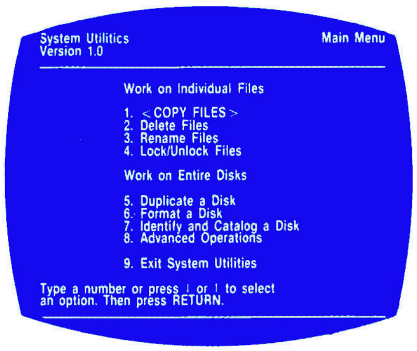
Copying Your Master Disk

The first thing you should do is copy your master disk. Like the DOS 3.3 COPYA utility, the ProDOS copy utility formats your disk automatically;



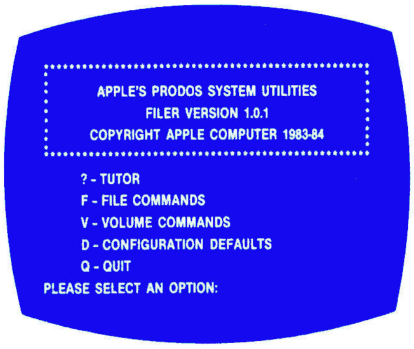
```
PRODOS USER'S DISK
COPYRIGHT APPLE COMPUTER, INC 1983

YOUR OPTIONS ARE:
      ? - TUTOR: PRODOS EXPLANATION
      F - PRODOS FILER (UTILITIES)
      C - DOS <-> PRODOS CONVERSION
      S - DISPLAY SLOT ASSIGNMENTS
      T - DISPLAY/SET TIME
      B - APPLESOFT BASIC
PLEASE SELECT ONE OF THE ABOVE
```

Figure 1. *User's Disk main menu.*



```
System Utilitics                          Main Menu
Version 1.0

            Work on Individual Files

            1.  <COPY FILES>
            2.  Delete Files
            3.  Rename Files
            4.  Lock/Unlock Files

            Work on Entire Disks

            5.  Duplicate a Disk
            6.  Format a Disk
            7.  Identify and Catalog a Disk
            8.  Advanced Operations

            9.  Exit System Utilities

Type a number or press ↓ or ↑ to select
an option. Then press RETURN.
```

Figure 2. *System Utilities main menu.*



```
APPLE'S PRODOS SYSTEM UTILITIES
FILER VERSION 1.0.1
COPYRIGHT APPLE COMPUTER 1983-84

      ? - TUTOR
      F - FILE COMMANDS
      V - VOLUME COMMANDS
      D - CONFIGURATION DEFAULTS
      Q - QUIT
PLEASE SELECT AN OPTION:
```

Figure 3. *User's Disk "Filer" menu.*

you don't need a formatted disk.

Here's the technique for Disk II and DuoDisk owners. (//c owners should skip to the section headed "Copying the System Utilities Disk.")

1) Insert the ProDOS User's Disk in drive 1 and start the system either by turning the computer on or by typing PR#6 and pressing the return key. The computer will display the main menu, shown in **Figure 1**, on the video screen.
2) From the main menu type F in upper- or lowercase to select the "Filer" utilities. The computer will display the "Filer" menu shown in **Figure 3**.
3) From the "Filer" menu, type V to select the "Volume" utilities. The computer will display the "Volume" menu shown in **Figure 4**.
4) From the "Volume" menu, type C to select the "Copy" utility. The computer will display the menu shown in **Figure 5**.
5) If you have two disk drives, place a blank disk in drive 2. Respond to the questions in the "Copy" menu by telling the computer to "Copy the volume in Slot 6, Drive 1 to the volume in Slot 6, Drive 2." Because these are the default values, you don't have to type the numbers; just press return and the computer will type the numbers for you.
6) If you have one disk drive, type in the numbers 6, 1, 6, and 1. Don't worry, the computer will tell you when to switch disks. The first three numbers are default values, so just press return to accept them. Type the final 1 before pressing return.
7) The "Copy" menu also allows you to specify a name for the new volume. Because you are making an exact copy of the User's Disk, simply press return. The computer will give the same name to the copy and display it in the screen area after the prompt "New Volume Name."
8) Press return again to accept the default name and start copying. The computer will first indicate that it is "Formatting" the blank disk. Then it will indicate that it is repeatedly "Reading" and "Writing" pieces of files from the old to the new disk. The in-use lights on the two drives will blink on and off accordingly. When it is done, the computer will return you to the "Copy" menu. For those with one disk drive, the computer will tell you when to switch disks.
9) Press the escape key twice to return to the "Filer" menu.
10) From the "Filer" menu, type Q to exit the menu. The computer will display the "Quit" menu and ask which program you want to load when you quit. It will suggest "BASIC.SYSTEM," which is the ProDOS-BASIC link.
11) Press the return key to accept the default. The computer will restart ProDOS and redisplay the User's Disk main menu. You have finished copying the disk.

Copying the System Utilities Disk

If you are an Apple //c owner, you employ a different method to copy your master disk.

1) Boot the System Utilities disk and wait until the menu shown in **Figure 2** appears.
2) Select the "Duplicate a Disk" option by typing 5 and hitting the return key.
3) From the next menu, hit the return key to choose the built-in drive as the location of your source disk.
4) From the next menu, hit return if you have one disk drive, type 2 and hit return if you have an external drive.
5) Make sure that System Utilities is in the internal drive and hit return.
6) If you have two drives, put a blank disk into the external one. If you have only one drive, be sure to follow the prompts that tell you when to insert the blank disk (the destination disk) and when to put System Utilities (the source disk) into the drive. When you reach the menu shown in **Figure 6**, select the default volume name "/UTILITIES" by hitting the return key.
7) The computer will now format the blank disk and copy the contents of the System Utilities disk onto it. Owners of one-drive systems will have to do a lot of disk swapping, as directed by the computer.
8) When the "Duplicating. . .Done!" message appears, hit the escape key to get back to the main menu.

Now that you have a copy of your master disk, whether it is Version 1.0.1 or 1.0.2, label it and store the original in a safe place. Use the copy from now on.

Formatting a Disk

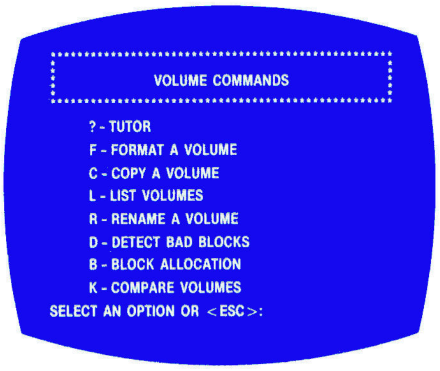
When you buy disks they are blank, but your Apple II can't use a blank disk. It must store information in a specific pattern on the disk so it knows where to find the information later. Therefore, before you can start to store information on a disk, you must *format* (initialize) it—that is, record a magnetic roadmap on the disk for the computer to follow.

In DOS 3.3, the INIT command formats the disk, but ProDOS doesn't have an INIT command (we will see why later in this series). Instead, your master disk contains a utility to format a blank disk.

It's good practice to format all the disks in a box as soon as you open it. That way you won't be caught in the middle of some important program without a formatted disk and without a way to create one.

To learn how to format a ProDOS disk, refer to "Beginner's Cookbook" on pages 110–111 of this month's *inCider*. It will tell you all you need to know.

ProDOS eases formatting by providing menus for you to follow, but the procedure is still not simple and not one you'll want to do often. The great-



```
VOLUME COMMANDS

    ? - TUTOR
    F - FORMAT A VOLUME
    C - COPY A VOLUME
    L - LIST VOLUMES
    R - RENAME A VOLUME
    D - DETECT BAD BLOCKS
    B - BLOCK ALLOCATION
    K - COMPARE VOLUMES
SELECT AN OPTION OR <ESC>:
```

Figure 4. *User's Disk "Volume" menu.*

er difficulty of formatting disks in ProDOS (compared to DOS 3.3) makes it all the more important that you keep several formatted disks available.

Other Volume Utilities

Formatting and copying are ProDOS volume (disk) utilities; there are other such utilities as well. Other Version 1.0.1 volume options include the following:

● *List Volumes.* Press the L key to list the slot number, drive number and volume name of all disks currently available to the computer. ProDOS automatically finds disk drives and determines the names of the disks in each.

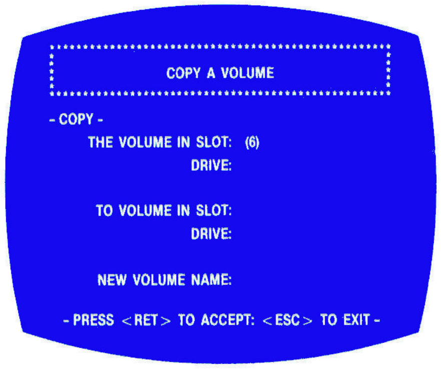
● *Rename a Volume.* Press the R key to display a menu for changing the name of a volume.

● *Detect Bad Blocks.* Press the D key to test the integrity of a disk. ProDOS will analyze each block on the disk and tell you which, if any, are "bad" (contain defects). If the disk is "good," the computer will display the message "0 BAD BLOCKS."

● *Block Allocation.* Press the B key to find the remaining capacity of a disk in any slot or drive. ProDOS will examine the directories on the disk you select to find the number of free (unused) and used blocks. As we will see in Part 4 of this series, the CAT and CATALOG commands also yield this information.

● *Compare Volumes.* Press the K key to make ProDOS compare two disks of your choice. ProDOS reads each block on each disk and compares them byte-by-byte to make doubly sure you have copied a disk correctly. If the two disks are identical, the computer will display the message "Compare Complete."

Additional Version 1.0.2 volume options are accessed by selecting num-



```
COPY A VOLUME

- COPY -
    THE VOLUME IN SLOT:  (6)
                  DRIVE:

     TO VOLUME IN SLOT:
                  DRIVE:

     NEW VOLUME NAME:

 - PRESS <RET> TO ACCEPT: <ESC> TO EXIT -
```

Figure 5. *User's Disk "Copy Volume" menu.*

ber 8 from the main menu. The options are similar to those described for Version 1.0.1.

File Maintenance

When you format and copy an entire disk or use any of the utilities just listed, you work from a volume menu. Below the volume level lie subdirectories and files. ProDOS also allows you to manipulate these levels.

Version 1.0.1 File Commands

Go to the "Filer" menu again. Type F to select the "File Commands." The computer will display the menu shown in **Figure 7**. You may select one of the following options:

● *List ProDOS Directory.* Type L to list the files in a directory. The computer will ask for the pathname. If the User's Disk is still in drive 1, type /USERS.DISK and press return. The computer will list the User's Disk files on the video screen:

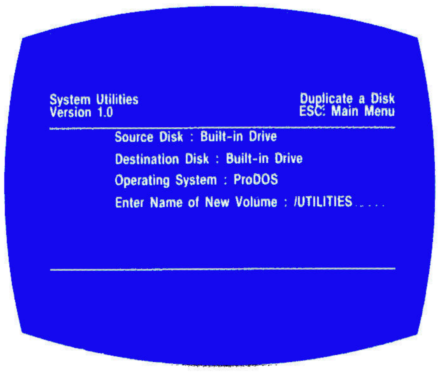
PRODOS
BASIC.SYSTEM
FILER
CONVERT
STARTUP
MOIRE
HYPNOSIS
ANIMALS

It will also list information about the files, but we will defer our discussion of this feature until Part 4.

● *Copy Files.* Type C to copy a file from one volume to another *or from one directory to another.* You must type the complete pathname of the source and destination files (and don't forget the leading slash!).

● *Delete Files.* Type D to delete a file from a directory. Once again, you will have to type the complete pathname of the file you want to delete. As with DOS 3.3, you cannot delete locked files (see "Alter Write Protection" below).

● *Compare Files.* Type K to compare



```
System Utilities                          Duplicate a Disk
Version 1.0                               ESC: Main Menu

          Source Disk : Built-in Drive
          Destination Disk : Built-in Drive
          Operating System : ProDOS
          Enter Name of New Volume : /UTILITIES.....
```

Figure 6. *System Utilities "Duplication" menu.*

two files byte-by-byte to make sure they match exactly. As with the "Compare" option in the "Volume Command" menu, you can use this utility to make sure you copied a file correctly.

● *Alter Write Protection.* Type A to lock unlocked files and vice versa.

● *Rename Files.* Type R to change the name of a file. The computer will ask you for the pathname under which the file now resides and the pathname under which you want it to reside. One caution here: You cannot rename a file from one directory or volume to another (you must use the "Copy File" option for that), so the volume name and subdirectory names must be the same in both pathnames.

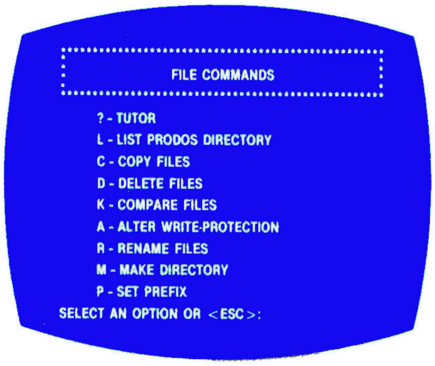
● *Make Directory.* Type M to create a new subdirectory. The computer will ask you to type the pathname. Unless you type the full pathname, the computer will assume you want to create a new directory on the startup disk (in this case, the ProDOS User's Disk). For example, to create a new subdirectory named TEST on the User's Disk, you may merely type TEST and press return. The computer will assume you want the directory on the User's Disk. You could also type /USERS.DISK/TEST and press return.

● *Set Prefix.* Type P to set a prefix. The prefix "teaches" the computer part of a pathname for it to automatically use until you change the prefix.

Version 1.0.2 File Commands

//c owners can also copy, delete, rename, lock and unlock, and list files from the main menu. The "Set Prefix" and "Create a Subdirectory" options are accessed from the "Advanced Operations" item of the main menu.

Let's work through a sample exercise that uses some ProDOS file utili-



```
..........................................
:              FILE COMMANDS             :
..........................................

        ? - TUTOR
        L - LIST PRODOS DIRECTORY
        C - COPY FILES
        D - DELETE FILES
        K - COMPARE FILES
        A - ALTER WRITE-PROTECTION
        R - RENAME FILES
        M - MAKE DIRECTORY
        P - SET PREFIX
SELECT AN OPTION OR <ESC>:
```

Figure 7. *User's Disk "File" menu.*

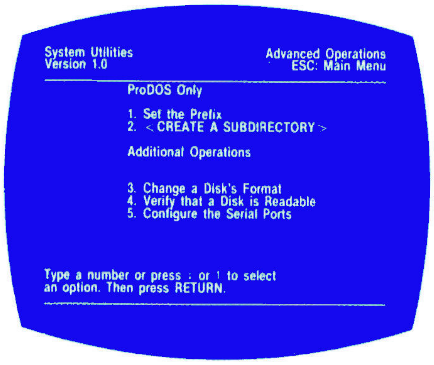
ties from the menus, first using Version 1.0.1 and then Version 1.0.2.

The ProDOS User's Disk contains three Applesoft BASIC programs: MOIRE, HYPNOSIS, and ANIMALS. We'll create a subdirectory and move those programs into it.

Here's how with Version 1.0.1:

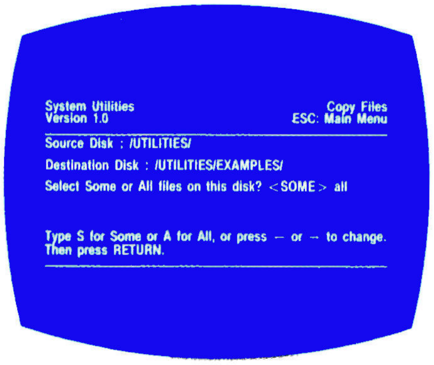
1) Boot your User's Disk (follow step 1 of the "Copying Your Master Disk" instructions.)
2) From the main menu, type F to select the "Filer" utilities. The computer will display the "Filer" menu (**Figure 3**).
3) From the "Filer" menu, type F again to select the "File" utilities. The computer will display the "File Commands" menu (**Figure 7**).
4) From the "File Commands" menu type M to select the "Make Directory" option. The computer will ask you for the pathname.
5) The menu header displays the current prefix, "/USERS.DISK/." Because we want to make a subdirectory in the same volume, we don't need to specify the prefix in the pathname. Type the subdirectory name EXAMPLES (without the leading slash, because that is part of the prefix already) and press return. The disk drive will turn on and off, then the computer will display the message "MAKE DIRECTORY COMPLETE," indicating the computer has added the new subdirectory /USERS.DISK/EXAMPLES on the volume USERS.DISK.
6) Press the escape key to return to the "File Commands" menu.
7) From the "File Commands" menu, type C to select the "Copy Files" option. The computer will display the "Copy Files" menu.
8) You must enter two pathnames: the pathname of the file you want to copy and the pathname of the file after you copy it. Note in the menu header that the prefix is still "/USERS.DISK/." For the first pathname, type MOIRE—the first file you will copy—and press return.
9) The computer will ask you to type the second pathname. Type EXAMPLES/MOIRE and press return twice. Drive 1 will come on, make some swooshing sounds, then go off. The computer will display the message "COPY COMPLETE." You have copied the file MOIRE from the volume /USERS.DISK/ to the subdirectory /USERS.DISK/EXAMPLES.
10) Repeat steps 8 and 9, first using HYPNOSIS and EXAMPLES/HYPNOSIS as the two pathnames, then ANIMALS and EXAMPLES/ANIMALS. You have copied all three files to the subdirectory EXAMPLES.
11) Press the escape key to return to the "File Commands" menu.
12) From the "File Commands" menu, type L to select the "List ProDOS Directory" option.
13) Type the pathname EXAMPLES (the prefix is still /USERS.DISK/). The computer will list the files in the subdirectory EXAMPLES: "MOIRE," "HYPNOSIS," and "ANIMALS."
14) Press the escape key to return to the "File Commands" menu.
15) Press escape again to return to the "Filer" menu.
16) Type Q to quit the "Filer" menu.
17) Press return to accept the computer's suggestion that you quit to "BASIC.SYSTEM." The computer will display the ProDOS User's Disk main menu.

See how simple ProDOS is to use from the built-in menus? You now have two copies of each of the three programs, one in the /USERS.DISK volume directory and one in the /USERS.DISK/EXAMPLES subdirectory. Try using the "Delete Files" option to delete the three files in the /USERS.DISK directory.



```
System Utilities                      Advanced Operations
Version 1.0                                ESC: Main Menu
---------------------------------------------------------
             ProDOS Only

             1. Set the Prefix
             2. < CREATE A SUBDIRECTORY >

             Additional Operations


             3. Change a Disk's Format
             4. Verify that a Disk is Readable
             5. Configure the Serial Ports



Type a number or press : or ! to select
an option. Then press RETURN.
---------------------------------------------------------
```

Figure 8. *System Utilities "Advanced Operations" menu.*



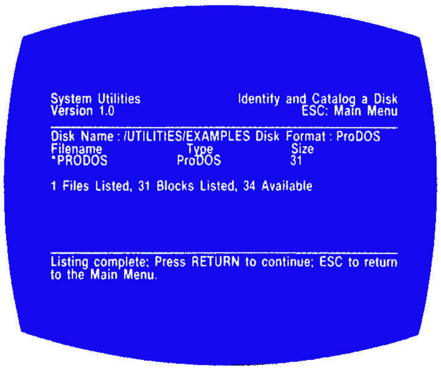
```
System Utilities                               Copy Files
Version 1.0                                ESC: Main Menu
---------------------------------------------------------
Source Disk : /UTILITIES/

Destination Disk : /UTILITIES/EXAMPLES/

Select Some or All files on this disk? <SOME> all


Type S for Some or A for All, or press — or → to change.
Then press RETURN.
---------------------------------------------------------
```

Figure 9. *System Utilities "File Copy" menu.*

Creating a Subdirectory with Version 1.0.2

Creating a subdirectory with ProDOS 1.0.2 differs from the procedure described above. Follow these steps:

1) Select "Advanced Operations" from the main menu.
2) From the next menu, shown in **Figure 8**, select "Create a Subdirectory."
3) From the menu after that, hit return to select the built-in drive.
4) You now have to supply a name for your new subdirectory. Type in EXAMPLES and hit return.
5) When the subdirectory has been created, hit the escape key twice to get back to the main menu.
6) Copying files into a subdirectory is a bit tricky in Version 1.0.2. From the main menu, select the "Copy Files" option.
7) From the next menu, type in 3 and hit the return key.
8) Hit return again to indicate that "/UTILITIES" will be the source of the files to be copied.
9) From the next menu, hit 3 and the return key.
10) You now have to enter the pathname of the subdirectory you will be copying to. Since the subdirectory is on the /UTILITIES volume, type in EXAMPLES and hit return.
11) Your screen should look like **Figure 9**. Hit return to get a list of the files on /UTILITIES. Note that your new subdirectory is on the list.
12) Mark ProDOS as the file to be copied by following the instructions at the bottom of the screen, then hit return.
13) When the computer prompts you to insert the destination disk, simply hit the return key. You are not copying ProDOS onto a new disk, but rather into a subdirectory on the same disk.
14) When you get the "Copying ProDOS. . .Done!" message, hit the escape key to return to the main menu.



```
System Utilities                    Identify and Catalog a Disk
Version 1.0                                  ESC: Main Menu

Disk Name : /UTILITIES/EXAMPLES Disk Format : ProDOS
Filename              Type             Size
*PRODOS               ProDOS           31

1 Files Listed, 31 Blocks Listed, 34 Available


Listing complete: Press RETURN to continue; ESC to return
to the Main Menu.
```

Figure 10. *System Utilities "Identify and Catalog" display.*

15) To verify that ProDOS has indeed been copied into the EXAMPLES subdirectory, hit 7 and the return key.
16) From the next menu, hit 3 and return.
17) Next, type in EXAMPLES and hit return.
18) Select the "Display" option by hitting return. You will see the screen shown in **Figure 10**, indicating that ProDOS has been copied into the EXAMPLES subdirectory. Hit the escape key to return to the main menu.

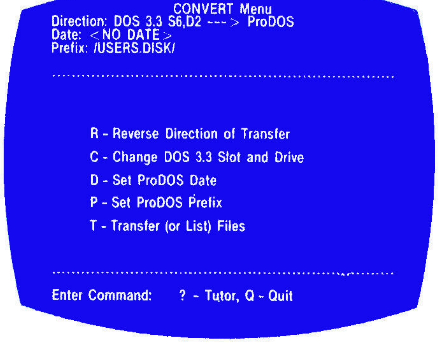
Converting DOS 3.3 Files

ProDOS and DOS 3.3 are incompatible. ProDOS includes a utility, however, that converts DOS 3.3 files to ProDOS disk format. The following instructions lead you step-by-step through the conversion. First, Version 1.0.1:

1) Go to the User's Disk main menu.
2) Type C to select the "DOS 3.3-ProDOS Conversion" utility. The computer will display the menu shown in **Figure 11**. The top line of the menu indicates the direction of the transfer. The default assumes a transfer from DOS 3.3 to ProDOS. The menu options allow you to:

- Change the direction of transfer (from ProDOS to DOS 3.3).
- Change the drive in which the DOS 3.3 disk is located.
- Reset the date.
- Set a ProDOS prefix.
- Start transferring files (assuming the above parameters are correct).

3) Type T to select the option to transfer files. The computer will ask you to type the name of the DOS 3.3 file you want to transfer.
4) Place the DOS 3.3 disk in drive 2 (if you have only one drive, wait for the computer to tell you when to switch disks), type the name of the file you want to transfer to ProDOS, and press return. If you want to transfer all the



```
                 CONVERT Menu
Direction: DOS 3.3 S6,D2 ---> ProDOS
Date: <NO DATE>
Prefix: /USERS.DISK/
..........................................................


      R - Reverse Direction of Transfer
      C - Change DOS 3.3 Slot and Drive
      D - Set ProDOS Date
      P - Set ProDOS Prefix
      T - Transfer (or List) Files


..........................................................
Enter Command:     ? - Tutor, Q - Quit
```

Figure 11. *User's Disk "Conversion" menu.*

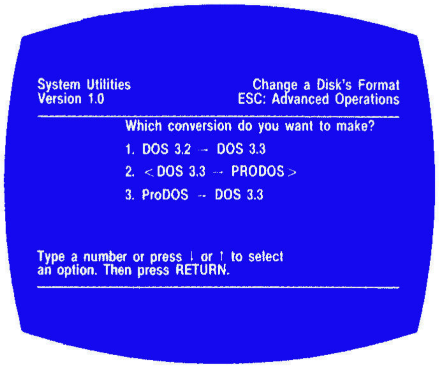
files on a DOS 3.3 disk to ProDOS, type an equals sign (=) and press return. ProDOS will tell you it is reading the file. After some disk-drive clunking, the computer will announce that it is done.
5) Press the escape key to return to the "Convert" menu.
6) Type Q to exit the "Convert" menu.
7) Press return to accept the computer's suggestion that you quit to "BASIC.SYSTEM." This completes the DOS 3.3-ProDOS file transfer.

You must exercise some caution using the DOS 3.3-ProDOS file transfer utility. First, the utility will not allow you to list the files on the DOS 3.3 disk. You must either remember the *exact* names of the DOS 3.3 files you want to transfer or use the equals sign to convert all DOS 3.3 files on the disk. If the DOS 3.3 disk is copy-protected, as many commercial disks are, you will not be able to convert the files.

In addition, the utility will not work equally well on all types of files. It must load the file to be transferred into the computer's memory before it can proceed. Applesoft programs and sequential text files transfer well. However, I have run into problems trying to transfer machine-language programs that occupy the same memory addresses as the Convert utility, because the utility protects itself from being destroyed. If you try such an "illegal" transfer, the computer will beep and stop the procedure. You cannot transfer random access text files.

Here is a summary of files you can and can't transfer:

Applesoft—yes
Sequential text—yes
Random access text—no
Binary—maybe



```
System Utilities                      Change a Disk's Format
Version 1.0                       ESC: Advanced Operations

              Which conversion do you want to make?
              1. DOS 3.2 → DOS 3.3
              2. <DOS 3.3 → PRODOS>
              3. ProDOS → DOS 3.3


Type a number or press ↓ or ↑ to select
an option. Then press RETURN.
```

Figure 12. *System Utilities "Conversion" menu.*

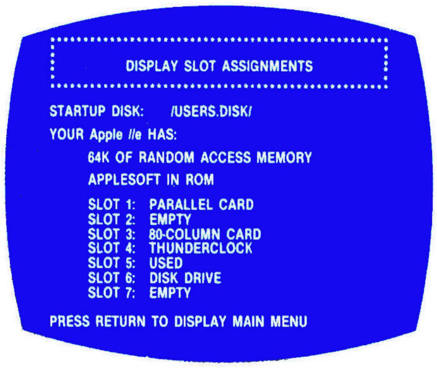
Integer—yes, but will not execute in ProDOS

Version 1.0.2
DOS-ProDOS Conversion

Follow these steps:

1) You access the conversion utility from "Advanced Operations," so type an 8 from the main menu and hit the return key.
2) From the next menu, type 3 and hit return.
3) From the menu after that, shown in **Figure 12**, select the type of conversion you want to make. To be consistent with the Version 1.0.1 example, type in 2 and hit return. If you don't have any DOS 3.3 files, you can abandon this exercise by hitting the escape key twice. You can't convert files you don't have!
4) If you're still with me, hit the return key again to select the internal drive.
5) Hit return once more.
6) You will now be prompted to insert the source disk into the drive. Make sure that it is a DOS 3.3 disk. Hit return.
7) When prompted, insert a blank disk into the drive and accept the default volume name by hitting return. ProDOS will format this disk as part of the conversion process.
8) Follow the prompts and swap disks as indicated by the computer. When you see the "Update Complete" message at the bottom of the screen, hit the escape key twice to return to the main menu. If you have a second drive, you will, of course, be spared the agony of disk swapping.

The caveats concerning conversions under Version 1.0.1 apply to Version 1.0.2, with one exception. Since the 1.0.2 conversion utility is located on the auxiliary bank of RAM, most of



```
DISPLAY SLOT ASSIGNMENTS

STARTUP DISK:   /USERS.DISK/
YOUR Apple //e HAS:
     64K OF RANDOM ACCESS MEMORY
     APPLESOFT IN ROM

     SLOT 1:  PARALLEL CARD
     SLOT 2:  EMPTY
     SLOT 3:  80-COLUMN CARD
     SLOT 4:  THUNDERCLOCK
     SLOT 5:  USED
     SLOT 6:  DISK DRIVE
     SLOT 7:  EMPTY

PRESS RETURN TO DISPLAY MAIN MENU
```

Figure 13. *User's Disk "Slot Assignments" display.*

your DOS 3.3 Binary files should convert to ProDOS without any trouble. Also, Version 1.0.2 converts a disk at a time, not a file at a time.

Other ProDOS Functions

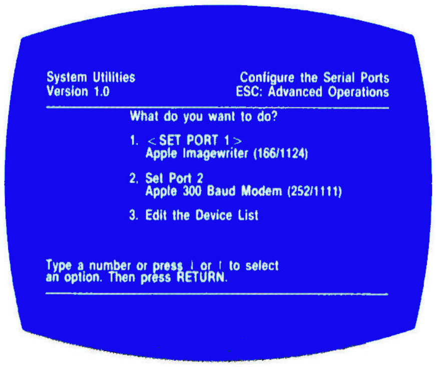
The User's Disk (Version 1.0.1) main menu includes three other functions. They include:

● *Display Slot Assignments.* Type S to tell ProDOS to look at each slot in your Apple and display the kind of card it contains in the format of **Figure 13**. This function is a bonus from ProDOS' need to find that information for its own use. ProDOS dates files whenever they are created or modified. When you tell ProDOS to look for a pathname, it must know where to find disk drives. So Apple built programs into ProDOS that test your computer to discover what kind it is, how much memory it has, and where the clock card (if you have one) and disk drives are. This utility merely displays this information to you.

● *Display/Set Time.* Type T to display or set the system time. This is the information ProDOS uses to date files (more on file dating in Part 4). If you have a clock card that uses the Mountain Computer/Thunderclock format, ProDOS will poll the clock to find out the time. When you change the ProDOS time, you *do not* change the clock time. If you don't have a clock card, you can set the time and date manually using this option. If you don't set a time and date, the computer will default to the value "NO DATE."

● *Applesoft BASIC.* Type B to transfer control of the system from the ProDOS User's Disk to *ProDOS* BASIC, a slightly enhanced version of Applesoft BASIC. The ProDOS-Applesoft link will be the primary subject of Part 3 of this series.

The System Utilities disk (Version 1.0.2) has some special features of its own. The most important of these is accessed via the "Advanced Operations" option of the main menu. Version 1.0.2 lets you configure the serial ports (**Figure 14**) of the //c so you can hook up just about any standard serial device to your computer (given the proper cabling). See chapter 4 of the System Utilities manual for a detailed explanation of this option.



```
System Utilities                Configure the Serial Ports
Version 1.0                     ESC: Advanced Operations
----------------------------------------------------------
                 What do you want to do?

                 1. < SET PORT 1 >
                    Apple Imagewriter (166/1124)

                 2. Set Port 2
                    Apple 300 Baud Modem (252/1111)

                 3. Edit the Device List


Type a number or press ↓ or ↑ to select
an option. Then press RETURN.
----------------------------------------------------------
```

Figure 14. *System Utilities "Serial Port Configuration" menu.*

In Part 2, we have examined the use of ProDOS from the built-in utility menus. For the casual user, these menus will probably provide all the capability needed. But you can also use ProDOS without the menus. The commands aren't that complicated once you understand the concepts of ProDOS we have just discussed. In upcoming months, we'll study how ProDOS and Applesoft BASIC fit together, some improvements Apple has made to BASIC, and how to use ProDOS from BASIC. I hope you will return next time to learn more about the power of ProDOS. ■

Lee Swoboda can't answer all questions individually, but if the answer is not lengthy and you include a stamped, self-addressed envelope, he will do his best to respond. Answers to other questions will appear in future articles or in the Letters column of inCider. *You can write to Lee Swoboda c/o Padapple Computer Consulting, 1451 NE Paulson Road, Poulsbo, WA 98370.*

Using ProDOS Part 3

Command Performance

by Lee Swoboda

This article continues a six-part series on ProDOS, Apple's new disk operating system (DOS). In Part 2 (see the April issue of *inCider*), I described how to use ProDOS from its built-in menus. In Parts 3 and 4 I'll examine the ProDOS commands, most often with hands-on examples of the command included. If you want to follow the examples, make a fresh copy of the ProDOS User's Disk or the System Utilities disk using the disk copying instructions in Part 2 of this series.

Get hands-on experience with ProDOS commands in both immediate and deferred modes.

ProDOS Commands

In the discussions of the ProDOS commands in the sidebar, the commands themselves appear in the headers in uppercase letters. Information you, the user, must provide is in lowercase. Optional parameters appear in italics. In all cases, quotes mean computer output and no-quotes means user input.

Most ProDOS commands can be used two ways. You can type the command directly from the Applesoft prompt (the immediate mode), or you may use it from within a BASIC program (the deferred mode). In the deferred mode, ProDOS follows the DOS 3.3 convention that requires DOS commands in a BASIC program to be used in a PRINT statement and include a control-D (ASCII 4).

For example, if you want to delete a file from the immediate mode, type DELETE and the appropriate pathname following the Applesoft prompt and press the return key. If you want your program to delete a file from the deferred mode, insert the following line in your Applesoft program with the appropriate line number and filename:

```
999 PRINT CHR$(4);"DELETE filename"
```

The CHR$(4) (ASCII 4) tells the computer that this is a ProDOS command, so the computer relinquishes control to ProDOS. The data between the quotes is then passed from Applesoft to ProDOS.

I will provide examples for using commands in both the immediate and deferred modes. If I show an example for only one mode, the command can be used only in that mode or is rarely used in the omitted mode.

In DOS 3.3, the computer "remembers" which disk drive it is using—the "current drive." If you want to switch to a different disk drive in DOS 3.3, you must tell the computer which one by

***Table 1.** A comparative listing of ProDOS and DOS 3.3 commands—those that function similarly under both operating systems; those that appear in DOS 3.3 but not in ProDOS; and those that appear in DOS 3.3 and have been improved in ProDOS.*

Similar	**Not in ProDOS**	**Improved in ProDOS**	
LOCK	FP	CATALOG	CLOSE
UNLOCK	INT	RUN	POSITION
LOAD	INIT	APPEND	READ
SAVE	MON	BLOAD	WRITE
RENAME	NOMON	BRUN	IN#
DELETE	MAXFILES	BSAVE	PR#
EXEC	VERIFY	OPEN	

***Table 2.** Comparison of ProDOS and DOS 3.3 command formats. Italic portions are optional; all the rest are mandatory.*

ProDOS Command Format	**DOS Command Format**
BLOAD pathname *,An,Bn,Ln,Txxx,Sn,Dn*	BLOAD filename,*An,Sn,Dn,Vn*
BLOAD pathname *,An,Bn,En,Txxx,Sn,Dn*	
BRUN pathname,*An,Bn,Ln,Sn,Dn*	BRUN filename,*An,Sn,Dn,Vn*
BRUN pathname,*An,Bn,En,Sn,Dn*	
BSAVE pathname,An,Ln *,Bn,Txxx,Sn,Dn*	BSAVE filename,An,Ln,*Sn,Dn,Vn*
BSAVE pathname,An,En *,Bn,Txxx,Sn,Dn*	
CAT *pathname,Sn,Dn*	- - -
CATALOG *pathname,Sn,Dn*	CATALOG,*Sn,Dn*
CHAIN pathname,*@n,Sn,Dn*	- - -
CLOSE *pathname*	CLOSE *filename,Sn,Dn*
CREATE pathname,*Txxx,Sn,Dn*	- - -
DELETE pathname,*Sn,Dn*	DELETE filename,*Sn,Dn,Vn*
EXEC pathname,*Fn,Sn,Dn*	EXEC filename,*Rn,Sn,Dn,Vn*
EXEC pathname,*Rn,Sn,Dn*	
FLUSH *pathname,Sn,Dn*	- - -
FRE	- - -
IN#n	IN#n
IN#An	
LOAD pathname,*Sn,Dn*	LOAD filename,*Sn,Dn,Vn*
LOCK pathname,*Sn,Dn*	LOCK filename,*Sn,Dn,Vn*
POSITION pathname,Fn	POSITION filename,*Rn*
POSITION pathname,Rn	
PREFIX *pathname,Sn,Dn*	- - -
PR#n	PR#n
PR#An	
PR#n,An	
RENAME path.1,path.2,*Sn,Dn*	RENAME file 1,file 2,*Sn,Dn,Vn*
RESTORE pathname,*Sn,Dn*	- - -
RUN pathname,*@n,Sn,Dn*	RUN filename,*Sn,Dn,Vn*
SAVE pathname,*Sn,Dn*	SAVE filename,*Sn,Dn,Vn*
STORE pathname,*Sn,Dn*	- - -
UNLOCK pathname,*Sn,Dn*	UNLOCK filename,*Sn,Dn,Vn*
-pathname,*Sn,Dn*	- - -
. . . . **Sequential Text File Commands**	
APPEND pathname,*Txxx,Sn,Dn*	APPEND filename,*Sn,Dn,Vn*
OPEN pathname,*Txxx,Sn,Dn*	OPEN filename,*Sn,Dn,Vn*
READ pathname,*Fn,Bn*	READ filename,*Bn*
WRITE pathname,*Fn,Bn*	WRITE filename,*Bn*
. . . . **Random-Access Text File Commands**	
APPEND pathname,*Ln,Sn,Dn*	- - -
OPEN pathname,Ln,*Sn,Dn*	OPEN filename,Ln,*Sn,Dn,Vn*
READ pathname,Rn,*Fn,Bn*	READ filename,Rn,*Bn*
WRITE pathname,Rn,*Fn,Bn*	WRITE filename,Rn,*Bn*

specifying the *slot* and *drive* parameters. In ProDOS, the *volume name* tells the computer which disk to use. When you type a pathname, the computer looks at each disk drive until it finds the disk with that name. (The computer "remembers" the name of any disk it has looked at, so it can return to that disk again without having to search.) In ProDOS the slot and drive numbers of a disk are superfluous. However, Apple has retained the slot and drive numbers as optional parameters in ProDOS commands, probably as a concession to compatibility with DOS 3.3. We won't discuss the use of slot and drive parameters because ProDOS doesn't need them.

You probably realize by now that DOS 3.3 and ProDOS are significantly different. Some DOS 3.3 commands are the same in ProDOS, but many are different. Even for those that are the same, you must observe ProDOS pathname conventions. For example, the format of the SAVE command in DOS 3.3 is:

SAVE filename,*Sn,Dn*

where the slot (,Sn) and drive (,Dn) numbers shown in italics are optional. In ProDOS, the format of the SAVE command is similar:

SAVE pathname,*Sn,Dn*

where the slot and drive numbers are optional. The filename must follow the DOS 3.3 rules for filenames and the pathname must follow the ProDOS rules for pathnames, which I discussed in Part 1.

Commands Similar in ProDOS and DOS 3.3

Seven ProDOS and DOS 3.3 commands are similar, except for the pathname distinction. They are listed in **Table 1.** Since the primary purpose of this series is to highlight the differences between DOS 3.3 and ProDOS, I'll discuss the uses of these common commands only briefly.

If you intend to follow the hands-on section with the description of each command, start your ProDOS User's Disk and type B from the main menu to go to the Applesoft BASIC prompt (]). If you are using ProDOS on a //c, boot the System Utilities disk and type 9 to exit System Utilities.

Table 2 summarizes the format of each ProDOS command and that of the equivalent command in DOS 3.3. The values following the command are parameters that amplify the command. **Table 3** defines these parameters and lists the range of their values. Parameters in italics are optional. Some commands have more than one format. The headers in the sidebar duplicate the command format in **Table 2.**

Deleted Commands

Not all ProDOS commands are similar to DOS 3.3. In fact, seven DOS 3.3 commands are not even available in ProDOS. You'll find them in **Table 1** also. If you are interested in the reasons Apple deleted these commands, see pages 201–202 of *BASIC Programming with ProDOS.*

Improved Commands

Apple improved 13 DOS 3.3 commands when they wrote ProDOS. Refer again to **Table 1.** You may still use all 13 the same way you did in DOS 3.3, but each has additional features that either enhance their usefulness or correct deficiencies in DOS 3.3. Part 4 of "The Compleat Text File Primer" in the June 1984 issue of *inCider* covered the file-access commands, beginning with APPEND, in more detail. Also refer to *BASIC Programming with ProDOS.*

In Part 3, we have studied the commands ProDOS shares with DOS 3.3. In Part 4, we will examine the *new* commands Apple has added to ProDOS. See you next month! ■

Lee Swoboda can't answer all questions individually. However, if the answer is not lengthy and you include a stamped, self-addressed envelope, he'll do his best to respond. Answers to other questions will appear in future articles or the inCider *Letters column. Lee Swoboda's address is Padapple Computer Consulting, 1451 NE Paulson Road, Poulsbo, WA 98370.*

Table 3. *Comparison of ProDOS and DOS 3.3 command optional parameters.*

		Range of n	
Description	**Syntax**	**ProDOS**	**DOS**
Slot number	,Sn	1 to 7	1 to 7
Drive number	,Dn	1 or 2	1 or 2
Field number	,Fn	0 to 65535	n/a
Record number	,Rn	0 to 65535*	0 to 32767
Number of bytes	,Bn	0 to 65534**	0 to 32767
Address in RAM	,An	0 to 65535	0 to 65535
Length in bytes	,Ln	1 to 65535	0 to 32767
End address in RAM	,En	1 to 65535	n/a
At line number	,@n	0 to 65535	n/a
Slot number***	n	0 to 7	0 to 7
File type	Txxx	see Table 4	n/a

* Rn is limited to 16 megabytes divided by record length (L) or 65535, whichever is smaller.

** Bn is limited to one less than the record length (L – 1) or 65534, whichever is smaller.

*** Used in the IN# and PR# commands.

Table 4. *Comparison of ProDOS and DOS 3.3 file types.*

Contents of File	**ProDOS Type**	**DOS Type**
Sequential text	TXT	T
Random access text	TXT	T
Binary	BIN	B
Applesoft BASIC program	BAS	A
Relocatable file	REL	R
Integer BASIC program	INT*	I
Integer BASIC variables	INV*	-
Directory	DIR	-
Applesoft variables	VAR	-
User defined	$F1 to $F8	-
ProDOS system file	SYS	-
ProDOS added command file	CMD	-
ProDOS reserved	$F9	-
ProDOS reserved	$C0 to $EF	-
Typeless file	$00	-
Bad block file	BAD	-

* Not executable under ProDOS

Commands Common to DOS 3.3 and ProDOS

APPEND pathname,*Txxx,Sn,Dn* (sequential)
APPEND pathname,*Ln,Sn,Dn* (random access)

ProDOS examples:
APPEND /PETS/DOGS/DACHSHUND
999 PRINT D$;"APPEND /PETS/DOGS/DOBERMAN,L37"

The DOS 3.3 APPEND command attaches new records to the end of sequential text files only. The ProDOS version attaches data to the end of any file type. If you don't specify the file type, the computer will assume a sequential-access text file. If you are appending a random-access text file, you must specify the length parameter so the computer will know how long to make each record it is adding.

The DOS 3.3 APPEND command has to load the entire text file into memory before it can append new data. This means DOS 3.3 APPEND has no speed advantage over reading a file record-by-record. The ProDOS APPEND command goes directly to the end of the file to add new records. This increases the speed of appending records significantly.

BLOAD pathname,*An,Bn,Ln,Txxx,Sn,Dn*
BLOAD pathname,*An,Bn,En,Txxx,Sn,Dn*

ProDOS examples:
BLOAD SAMPLE,A$300,E$31B
999 PRINT D$;"BLOAD SAMPLE,A768,L23"

The BLOAD command loads, but does not execute, a binary file. The ProDOS version has three improvements from DOS 3.3:

● The BLOAD command will load the binary image (the way it appears in the computer's memory) of any file by specifying a type parameter. If you don't specify a type, the computer assumes a BIN file (see **Table 4**). In DOS 3.3, the only file you can BLOAD is a B-type (binary) file.
● The BLOAD command loads part of a binary file by specifying either the starting-address and length parameters or end-address parameters. If you don't specify these values, the computer will load the entire binary file in the location from which it was originally BSAVEd.
● The BLOAD command allows you to define the file by either the length or the end address. DOS 3.3 allows only the starting address and length parameters to define a binary file.

BRUN pathname,*An,Bn,Ln,Sn,Dn*
BRUN pathname,*An,Bn,En,Sn,Dn*

ProDOS examples:
BRUN /MAIN.DISK/SAMPLE
999 PRINT D$;"BRUN /MAIN.DISK/SAMPLE"

The ProDOS BRUN command loads and executes a binary (machine-language) program. Like BLOAD, BRUN can execute only part of a binary program. It has the same parameter options as the BLOAD command, except the type parameter. Unlike BLOAD, BRUN will execute only actual machine-language programs.

BSAVE pathname,An,Ln,*Bn,Txxx,Sn,Dn*
BSAVE pathname,An,En,*Bn,Txxx,Sn,Dn*

ProDOS examples:
BSAVE SAMPLE,A$300,L$1B
999 PRINT D$;"BSAVE SAMPLE,A768,E785"

The BSAVE command transfers the binary image of memory to the disk. The conditions and parameters are the same as for the BLOAD command.

CATALOG *pathname,Sn,Dn*

CATALOG is similar to CAT, except that CATALOG prints 80 columns of information instead of the 40 columns CAT prints. CATALOG provides the following information for each file in the pathname:

● Write-protect status
● Filename
● File type (see **Table 4**)
● Number of 512-byte disk blocks the file occupies
● Date and time the file was last modified
● Date the file was created
● The ENDFILE value of the file (the amount of disk space it takes)
● The file SUBTYPE (the memory address from which a BIN [binary] file was saved or the size of each element in a TXT [text] file)

If you have a 40-column screen, the computer will display the information on two lines per file. If you have an 80-column display card, the information will fit on one line. ProDOS doesn't switch the display to the 80-column mode; you have to do that yourself. (Type PR#3 and press the return key. To return to the 40-column mode, press open apple-control-reset on the Apple //e or control-reset on the II Plus.)

CLOSE *pathname*

ProDOS examples:
CLOSE
CLOSE /PETS/DOGS/DACHSHUND
999 PRINT D$;"CLOSE /PETS/DOGS/DACHSHUND"

The syntax of the CLOSE command is the same as in DOS 3.3. The command will close the file with the specified pathname. If you do not specify a pathname, the computer will close all open files. This is true in both DOS 3.3 and ProDOS.

ProDOS does not automatically close all open files when the program ends, as DOS 3.3 does. In ProDOS, you *must* close all open files before you exit a BASIC program. ProDOS flushes the file buffers of unwritten data when you close a file, so this requirement to close all open files is a precaution to assure all data is saved.

DELETE pathname,*Sn,Dn*

ProDOS examples:
DELETE /PETS/DOGS/DACHSHUND
999 PRINT D$;"DELETE /PETS/DOGS/DACHSHUND"

The DELETE command removes the specified file's listing from the disk directory so the computer "forgets" the file is there. However, it does not actually

continued

destroy the information in the file.

Because DOS 3.3 has only one level of directory organization, the DELETE command erases only files. The DELETE command in ProDOS will erase files or subdirectories (DIR files). However, to protect the user, ProDOS automatically locks DIR files, so you cannot delete a subdirectory while it contains any files. You must delete all the files in the subdirectory first.

Hands on:

From the BASIC prompt, type DELETE MOIRE.PATTERN and press return. When the disk drive stops running, type CAT and press return. The list of files now contains only "MOIRE".

EXEC pathname,*Fn*,*Sn*,*Dn*
EXEC pathname,*Rn*,*Sn*,*Dn*

ProDOS examples:
```
EXEC SAMPLE,F3
999 PRINT D$;"EXEC SAMPLE"
```

The EXEC command loads a sequential text file from disk to the computer and executes the file as a BASIC program. Obviously, the records in the EXEC file must be in the same format as a line in an Applesoft program or a ProDOS command.

Discussing the use of the EXEC command is beyond the scope of this series. If you want more information, see Chapter 8 of the *BASIC Programming with ProDOS* manual, which is included with the ProDOS BASIC Programmer's Kit (Apple product A202037).

IN#n, IN#An, PR#n, PR#An, and PR#n,An

ProDOS example:
```
999 PRINT D$;"PR#1"
```

The IN# and PR# commands control the peripheral slot from which the computer will receive input (IN#) or to which it will send output (PR#). In DOS 3.3 you can reference only peripheral slots 0–7. In ProDOS you can also use IN# and PR# to reference machine-language routines in memory. For example, the command PR# A$300 sends output to a machine-language program at memory address 300 (hexadecimal).

LOAD pathname,*Sn*,*Dn*

ProDOS example:
```
LOAD /PETS/DOGS/DACHSHUND
```

The LOAD command moves a BASIC program from disk to memory, but does not start the program running. This command is useful when you are writing BASIC programs. It allows you to load an incomplete program. LOAD works only with Applesoft BASIC files.

Hands on:

From the BASIC prompt, type LOAD MOIRE and press return. The computer will load the program into memory, but not execute it. List the program (type LIST and press return) to prove to yourself the computer did load the program from disk.

LOCK pathname,*Sn*,*Dn*
UNLOCK pathname,*Sn*,*Dn*

ProDOS examples:
```
LOCK /PETS/DOGS/DACHSHUND
999 PRINT D$;"UNLOCK /PETS/DOGS/DACHSHUND"
```

The LOCK command allows you to protect any file from deletion. UNLOCK removes the protection.

Hands on:

From the Applesoft BASIC prompt (]), type CAT and press return. (I'll discuss the CAT command later in this series.) The computer will list the files on the User's Disk on the video screen. Note that all the file listings have an asterisk (*) in the left column. This means they are all locked so you cannot modify them. Now type UNLOCK MOIRE (you don't need the volume name because the prefix is "/USERS.DISK/" already) and press return. When the disk drive goes off, type CAT again. This time the file "MOIRE" doesn't have an asterisk. You may lock "MOIRE" again by typing LOCK MOIRE, but don't actually do it because we need it unlocked for some later examples.

OPEN pathname,*Txxx*,*Sn*,*Dn* (sequential)
OPEN pathname,*Ln*,*Sn*,*Dn* (random access)

ProDOS examples:
```
999 PRINT D$;"OPEN /PETS/DOGS/DACHSHUND"
999 PRINT D$;"OPEN /PETS/DOGS/DACHSHUND,L34"
```

The OPEN command sets up a space (buffer) in memory to temporarily store data coming from or going to a file on disk. In DOS 3.3 you can open only a text file. In ProDOS you can open any file by specifying the type parameter. If you don't specify a file type, the computer will assume a text file. The length parameter is mandatory to specify for random-access text files when you first create the files. Unlike DOS 3.3, you cannot READ from or WRITE to an unopened file. Because ProDOS doesn't set up a file buffer until it opens a file, you must open a file before you read or write.

The OPEN command opens both sequential and random-access text files. Unless you specify an L (record length) parameter when you open a random-access text file, ProDOS will assume the record length is the same as when you created the file.

POSITION pathname,Fn
POSITION pathname,Rn

ProDOS example:
```
998 PRINT D$;"OPEN /PETS/DOGS/DACHSHUND"
999 PRINT D$;"POSITION /PETS/DOGS/DACHS-
HUND,R13"
```

The POSITION command moves you to a specified record number in a file without having to READ the preceding records. Since the R and F parameters (record and field parameters, see **Table 3**) for the READ and WRITE commands serve exactly the same function as the POSITION command, the POSITION command is superfluous. Apple included it in ProDOS to maintain compatibility with DOS 3.3.

READ pathname,*Fn*,*Bn* (sequential)
READ pathname,Rn,*Fn*,*Bn* (random access)

ProDOS example:
```
998 PRINT D$;"OPEN /PETS/DOGS/DACHSHUND"
999 PRINT D$;"READ /PETS/DOGS/DACHSHUND,F13"
```

The READ command allows you to access the information in either a sequential or random-access text

continued

(TXT) file (using the INPUT command). If you are accessing a random-access text file, the R parameter is mandatory.

The ProDOS READ command adds the R, F, and B parameters to allow you to start reading the file at a particular record, field, or byte within a record. This option makes the POSITION command superfluous.

RENAME pathname.1,pathname.2,*Sn*,*Dn*

ProDOS example:
```
RENAME DOGS/DACHSHUND,DOGS/WEENIE.DOG
```

The RENAME command allows you to change the name of a file. You cannot use RENAME to move files between directories or subdirectories. Therefore, *pathname.1* and *pathname.2* must have the same prefix; only the filename can be different. You cannot rename a locked file.

Hands on:

From the BASIC prompt, type RENAME MOIRE.1, MOIRE.PATTERN and press return. When the disk drive stops running, type CAT and press return. The computer will display a list of files that includes "MOIRE" and "MOIRE.PATTERN", but not "MOIRE.1".

RUN pathname,*@n*,*Sn*,*Dn*

ProDOS examples:
```
RUN /PETS/DOGS/DACHSHUND
999 PRINT D$;"RUN /PETS/DOGS/DACHSHUND"
```

The RUN command loads and executes a BASIC program. The ProDOS RUN command allows you to specify a line number where program execution will begin. If you do not specify a line number, execution begins at the first line.

SAVE pathname,*Sn*,*Dn*

ProDOS example:
```
SAVE /PETS/DOGS/DACHSHUND
```

The SAVE command transfers the current Applesoft program from memory to disk. It overwrites any existing program with the same pathname. You cannot SAVE a program if a locked file of the same name already exists.

Hands on:

From the BASIC prompt, type SAVE MOIRE.1 and press return. When the disk drive stops running, type CAT and press return. The computer will list the files on the User's Disk, now including both "MOIRE" and "MOIRE.1".

WRITE pathname,*Fn*,*Bn* (sequential)
WRITE pathname,Rn,*Fn*,*Bn* (random access)

ProDOS example:
```
998 PRINT D$;"OPEN /PETS/DOGS/DACHSHUND"
999 PRINT D$;"WRITE /PETS/DOGS/DACHSHUND"
```

The WRITE command prepares a text (TXT) file to save information in. In ProDOS, the WRITE command, like the READ command, allows you to specify the record number, field number, and byte where you want to begin. If you don't specify these parameters, the computer will start writing at the beginning of the file.

Using ProDOS Part 4

A New Order

by Lee Swoboda

Nine commands unique to ProDOS make life easier for the BASIC programmer.

Last month you learned about the commands ProDOS shares with DOS 3.3. To work its magic, Apple has also added nine new commands to ProDOS. They are listed in **Table 1**.

There are a number of things to keep in mind as you read the descriptions of the commands. First, in the model of the command format (in bold type), the commands themselves appear in uppercase letters, and information you, the user, must input is in lowercase. Information in italics is optional for you to enter. (**Table 2** defines these parameters and lists the range of their values.) Examples without line numbers are in immediate mode; all other examples are in deferred mode.

Table 1. Commands new to ProDOS.

CAT	FRE
CHAIN	PREFIX
CREATE	RESTORE
DASH (–)	STORE
FLUSH	

In the discussion of each command, you must provide all the information including the part in quotation marks. This latter information tells ProDOS what to do and what pathname to use. When used in a program, it won't appear on the screen. Following the discussion is a "hands-on" section which provides an example for you to try.

CAT *pathname,Sn,Dn*
ProDOS examples:
CAT
CAT /volumename/subdirectory
999 PRINT D$;"CAT"

Apple claims CAT is a new command, but it actually performs the same function as CATALOG does in DOS 3.3: It prints a 40-column list of the disk's files to the video screen. (See Part 3.)

The CAT display, however, doesn't look quite like the CATALOG display. The screen CAT produces is similar to the **Figure** and includes:

- An asterisk in the left column if the file is locked.
- The file's name.
- A three-letter abbreviation of the file's type (see **Table 3**).
- The number of 512-byte blocks the file occupies on the disk.
- The date the file was last modified (in mm/dd/yy format).
- The total number of 512-byte blocks

used on the entire disk (not just the current directory) and the number still available.

If you use the CAT command without a pathname, the computer will list the files in the default pathname—your "boot" drive, probably drive 1—unless you've changed it with the PREFIX command. If you do specify a pathname, the computer will list the files in that path. Specifying just the volume directory in the pathname makes CAT list only the subdirectories (DIR files) in that directory, but including a subdirectory in the pathname makes CAT list only the files in that subdirectory.

The CAT command will let you list all the files on the disk only if you don't have any subdirectories, so you must design your disk directories thoughtfully. If your directory structure is complex, you may find yourself CAT-ing through every directory trying to find a "lost" file. Conversely, if you've designed logical directories, you won't have to scan through several screens of files, as with DOS 3.3, to find the one you want; merely CAT the appropriate subdirectory. (See "More than Meets the Eye" by Viktor Rubenfeld, April 1985, p. 32.)

CHAIN pathname,*@n,Sn,Dn*
ProDOS example:
999 PRINT D$;"CHAIN /PETS/DOGS/
DOBERMAN"

When an Applesoft program ends, its program variables remain in memory. When the next Applesoft program begins execution, it destroys these variables. The CHAIN command saves the names and values of variables generated by one program for use in another. CHAIN combines the functions of STORE, RUN, and RESTORE.

Hands on:
From the BASIC prompt, type NEW and press the return key, then type the following BASIC program:

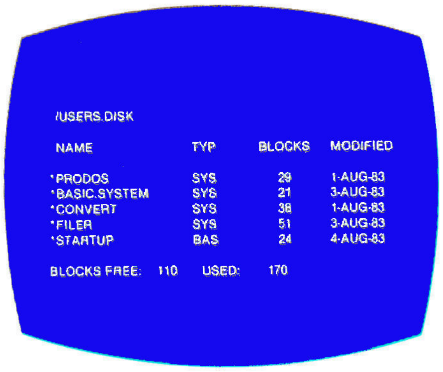
```
10 PRINT A
20 PRINT A$
30 END
```

Save the program as ETC, then type:

```
10 D$ = CHR$(4)
20 A = 1
30 A$ = "Hello"
40 PRINT D$;"CHAIN ETC"
```

Now type RUN and press the return key. The computer will print the value "1" and the string "Hello" on the screen. The second example above generated the values, then passed them to ETC using the CHAIN command. Type LIST and press return to

***Figure.** ProDOS CAT command screen.*



```
/USERS.DISK

NAME              TYP      BLOCKS   MODIFIED

*PRODOS           SYS        29     1-AUG-83
*BASIC.SYSTEM     SYS        21     3-AUG-83
*CONVERT          SYS        38     1-AUG-83
*FILER            SYS        51     3-AUG-83
*STARTUP          BAS        24     4-AUG-83

BLOCKS FREE:   110    USED:    170
```

verify that the first program executed the second. The CHAIN function is useful when you are constructing a program in modules; one module can end by chaining another module and passing its variables along.

CREATE pathname,*Txxx,Sn,Dn*
ProDOS examples:
CREATE /PETS/DOGS/DACHSHUND
999 PRINT D$;"CREATE /PETS/DOGS/
DACHSHUND"

ProDOS uses the CREATE command to establish a file. Normally you'd use it to establish new subdirectories, but you may establish any of the file types listed in **Table 3** by adding the type parameter to the command. You compose the type parameter from the letter T plus the three-character file type. If you don't specify a type, ProDOS will assume the default TDIR and establish a directory file.

When you use the CREATE command to establish a subdirectory, ProDOS establishes a valid DIR file in the directory, inserts all the appropriate information (dates, times, size) in the file, and then locks the file.

Just because you can establish file types other than DIR with CREATE doesn't mean you can do much with the file. ProDOS treats established files as sequential text files (which is what a DIR file is). So, if you CREATE a BAS file, when you try to load or run it, you will get an "Out of Data" error because there is nothing in the file. What the CREATE command does is reserve space in the directory.

Hands on:
From the BASIC prompt, type CREATE USERS.DISK/TEST (or CREATE UTILITIES/TEST if you're using the //c's version 1.0.2), and press the return key. When the drive stops running, CAT the disk. The computer will list the new file "TEST", which is a DIR (directory) file.

– pathname,*Sn,Dn*
ProDOS examples:
– /PETS/DOGS/DACHSHUND
999 PRINT D$;"– /PETS/DOGS/
DACHSHUND"

The DASH command (the symbol –) is a substitute and abbreviation for RUN, BRUN, and EXEC. The computer will look at the file type and select the correct command: RUN for file type BAS, BRUN for file type BIN, and EXEC for file type TXT.

Hands on:
From the BASIC prompt, type – STARTUP and press the return key to execute the start-up program.

FLUSH *pathname*
ProDOS examples:
FLUSH /PETS/DOGS/DACHSHUND
999 PRINT D$;"FLUSH /PETS/DOGS/
DACHSHUND"

The FLUSH command performs the rather gauche task that its name suggests: It transfers data in a file's buffer

Continued on p. 42.

***Table 2.** Optional parameters of ProDOS commands.*

Description	**Syntax**	**Range of n**
Slot number	,Sn	1 to 7
Drive number	,Dn	1 or 2
Field number	,Fn	0 to 65535
Record number	,Rn	0 to 65535*
Number of bytes	,Bn	0 to 65534**
Address in RAM	,An	0 to 65535
Length in bytes	,Ln	1 to 65535
End address in RAM	,En	1 to 65535
At line number	,@n	0 to 65535
Slot number***	n	0 to 7
File type	Txxx	see Table 3

* Rn is limited to 16 megabytes divided by record length (L) or 65535, whichever is smaller.

** Bn is limited to one less than the record length (L – 1) or 65534, whichever is smaller.

*** Used in the IN# and PR# commands.

Table 3. *ProDOS file types.*

Contents of File	Type
Sequential text	TXT
Random access text	TXT
Binary	BIN
Applesoft BASIC program	BAS
Relocatable file	REL
Integer BASIC program	INT*
Integer BASIC variables	INV*
Directory	DIR
Applesoft variables	VAR
User defined	$F1 to $F8
ProDOS system file	SYS
ProDOS added command file	CMD
ProDOS reserved	$F9
ProDOS reserved	$C0 to $EF
Typeless file	$00
Bad block file	BAD

* Not executable under ProDOS

Continued from p. 38.

to disk, or "flushes" the buffer. FLUSH is similar to a partial WRITE command, which writes only unsaved data to the file. If your program may stop unexpectedly, FLUSHing avoids losing data that hasn't been written to a file yet. For example, perhaps your program waits to write a series of new records to a file until after all the records have been input. You might want to FLUSH the file after you enter each record to ensure that a data-entry error on one record won't destroy all the others.

The CLOSE command also does a FLUSH to clear the file buffer before it closes a file.

FRE

ProDOS example:
```
10 PRINT D$;"FRE"
```

As Applesoft creates (concatenates) new strings from a collection of old ones, it leaves the old ones in memory. Eventually, free memory fills with "garbage" consisting of pieces of old strings. When that happens, the computer automatically "collects" the garbage—it gets rid of it—and you temporarily lose control of the keyboard. This process may take several minutes, and, of course, Murphy's Law dictates that this situation occurs at the most inconvenient point in the program. The Applesoft FRE command lets you control when your computer collects garbage, but it doesn't affect how long the collecting takes. You can only divide a long clean-up process into smaller pieces.

ProDOS's new FRE garbage-collection procedure is much faster than Applesoft's. The Applesoft FRE still works, but use the ProDOS example above to speed garbage collection.

PREFIX *pathname,Sn,Dn*

ProDOS examples:
```
PREFIX /USERS.DISK/
999 PRINT D$;"PREFIX /USERS.DISK/"
```

The PREFIX command "teaches" the computer part of a pathname so you won't have to type the entire pathname each time you want to refer to a file. If all the files you want to select are in a particular subdirectory, set the prefix to the volume name and subdirectory. You may then refer to the files by typing only their file name; the computer automatically adds the prefix.

Hands on:

From the BASIC prompt, type PREFIX /USERS.DISK/TEST (or PREFIX /UTILITIES/TEST for the //c) and press the return key. Type CAT and press return again. Because you didn't specify a pathname, the computer will list the files in the directory with the prefix name. There aren't any files in the subdirectory /TEST, so the computer won't list any.

STORE pathname,*Sn,Dn*

ProDOS examples:
```
STORE /PETS/DOGS/DACHS
999 PRINT D$;"STORE /PETS/DOGS/
    DACHS"
```

Normally, when you end a BASIC program, any variables the program used disappear into oblivion, unless you write them to a text file. STORE saves the names and current values of these variables in a special file (named "DACHS" in the example above), without your having to save each one individually. This special file is of the type VAR (see **Table 3**).

Hands on:

From the BASIC prompt, type NEW and press the return key, then type:
```
10 D$ = CHR$(4)
20 A = 1
30 A$ = "Hello"
40 PRINT D$;"STORE VARIABLES"
```

Type RUN and press the return key. When the drive stops running, type CAT and press return again. The computer lists the files in the subdirectory TEST, which now includes VARIABLES.

RESTORE pathname,*Sn,Dn*

ProDOS example:
```
RESTORE /PETS/DOGS/DACHS
PRINT D$;"RESTORE /PETS/DOGS/
    DACHS"
```

RESTORE recovers variable names and values that STORE saved.

Hands on:

To recover the variables STORE saved, type:
```
10 D$ = CHR$(4)
20 PRINT D$;"RESTORE VARIABLES"
30 PRINT A
40 PRINT A$
```

Type RUN and press the return key. The computer will print a "1" and "Hello" on the screen, indicating it has "remembered" both the name and value of the two variables in the STORE example.

COMING UP

Parts 5 and 6 of this series will delve more deeply into using ProDOS commands in BASIC programs, including the critical issue of making DOS 3.3 BASIC programs run under ProDOS. I hope you'll join me. ■

Lee Swoboda can't answer all your questions individually. However, if the answer is not lengthy and you include a stamped, self-addressed envelope, he'll do his best to respond. He'll include answers to other pertinent questions in future articles or in inCider's *Letters column. You can write to Lee Swoboda at Padapple Computer Consulting, 1451 NE Paulson Road, Poulsbo, WA 98370.*

Using ProDOS Part 5

The ProDOS–BASIC Connection

by Lee Swoboda

You've learned the commands—now discover how ProDOS works in an Applesoft BASIC program.

Now that you're familiar with the commands ProDOS offers and what they do, I'll explain how ProDOS works in an Applesoft BASIC program. When you turn on your Apple, it automatically loads your operating system from the disk, stores it in memory, and connects it to other functions of the computer. If that operating system is DOS 3.3, the Applesoft prompt (]) appears when this process is complete. ProDOS, however, doesn't automatically connect to BASIC; you must provide the system file—BASIC.SYSTEM—to connect the two.

ProDOS also differs from DOS 3.3 in the way it's stored in memory. (**Figure 1** compares the two for a 48K Apple II Plus.) After loading DOS 3.3, the computer stores it in a fixed location at the top of available memory—just above memory location 38400. Whenever a BASIC program uses a DOS 3.3 command (signified by the control-D in a PRINT statement), the computer goes to this location and looks at the table containing all DOS 3.3 commands. On the other hand, the only part of ProDOS that must be at a specific spot in memory is the table of parameters, the system global page (labeled ProDOS Table in **Figure 1**), which tells the computer where to find ProDOS.

Apple recommends a 64K configuration for ProDOS since this operating system has 7.5K more overhead (memory devoted to the computer's use) than DOS 3.3. If you're using ProDOS with a 48K Apple II Plus, ProDOS must reside just below the I/O buffers—the same place DOS 3.3 would reside. Since ProDOS is 1.5K bytes larger than DOS 3.3 and reserves the bottom 6K of memory for loading ProDOS or a system program, only 28K remains to hold BASIC programs (compared with 36K with DOS 3.3).

ProDOS automatically determines the type of computer you're using and how much memory is available. If you have a //e, a //c, or a II Plus, ProDOS locates itself in the 16K RAM card area of memory and updates its parameter table accordingly. **Figure 2** shows the memory configuration for a 64K Apple (48K Apple II Plus and 16K RAM card, //e, or //c). Because of the fixed location of DOS 3.3, however, even if you have an Apple //e, a //c, or a 64K Apple II Plus, DOS 3.3 can't easily take advantage of the additional 16K of available memory.

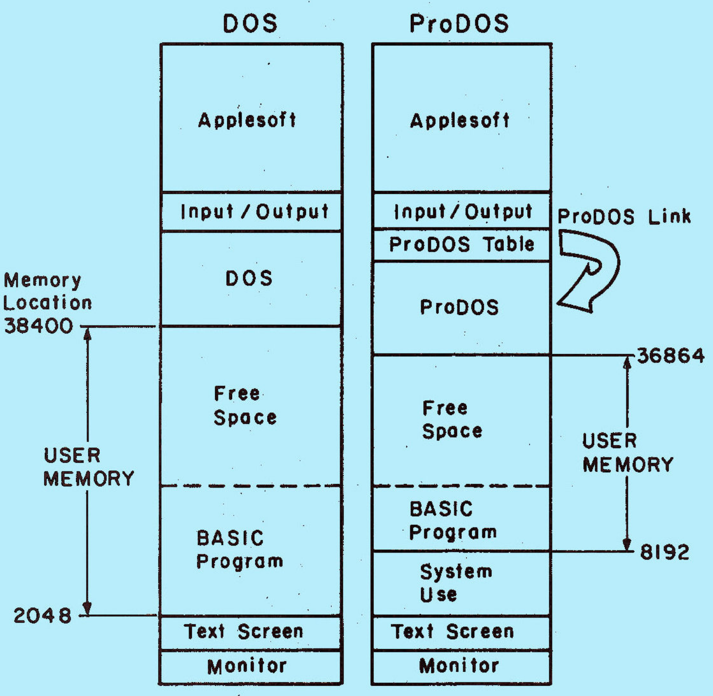


Figure 1. *DOS and ProDOS memory maps of a 48K Apple II Plus.*

Setting up a BASIC Disk

A ProDOS disk must have three characteristics: It must be formatted—magnetically on the disk; it must contain the file PRODOS (on which the ProDOS utilities are recorded); and it must contain a system file. (This file is BASIC.SYSTEM if you're using BASIC.)

To format a BASIC disk, follow the appropriate procedure in Beginner's Cookbook (p. 94). Format the blank disk using /PLAYING as the volume name. Your disk will boot (start up) in "ProDOS BASIC"—Applesoft BASIC with the enhancements discussed in Part 4 of this series. This formatting process is a more complicated version of the DOS 3.3 INIT command, but it's worth the effort to have the extra features ProDOS offers. To format additional disks, you merely need to use the Volume Copy Utility. I recommend you use your newly created disk as a master for future BASIC boot disks.

Now, put /PLAYING in drive 1, type PR#6, and press the return key. The disk boots your system and displays "ProDOS BASIC 1.0" on the screen. When you first start ProDOS, it looks for a program named STARTUP. Since /PLAYING doesn't have such a program, ProDOS displays the BASIC prompt (]). Normally, you'll want the computer to run a program automatically when you first start it, rather than dump you in BASIC. To create a simple start-up program, from the BASIC prompt, type:

```
10 PRINT CHR$(4);"CAT"
```

and press the return key. Type SAVE STARTUP and press the return key. Now, whenever you boot /PLAYING, the computer will automatically display a 40-column catalog before depositing you in BASIC.

From the BASIC prompt, type CAT and press the return key. The computer lists the three files on your disk: PRODOS, BASIC.SYSTEM, and STARTUP. These three files are fundamental to any disk that uses ProDOS from BASIC. If you write your own BASIC program, name it STARTUP so the computer will run your program every time you start the disk.

Starting a BASIC program in ProDOS is similar to starting one in DOS 3.3. You will need to exercise special care, however, with some ProDOS commands. (See **Table 1**.)

Changes to Applesoft

Ten Applesoft commands—HIMEM, HGR, HGR2, TEXT, INPUT, IN#, PR#, TRACE, NOTRACE, and FRE—behave differently depending on which operating system (ProDOS or DOS 3.3) you're using. **Table 2** briefly discusses each command. For more information, refer to *BASIC Programming with ProDOS*.

Converting Applesoft Programs to ProDOS

The primary reason to convert any program to ProDOS is to avoid switching between two operating systems. (Long-time Apple owners may remember switching between DOS 3.2 and 3.3.) A more significant reason is that for certain types of BASIC programs ProDOS offers advantages over DOS 3.3:

● Programs that use sequential text files. ProDOS reads, writes, and appends text files much more rapidly than DOS 3.3.

Figure 2. *ProDOS memory map of a 64K Apple.*

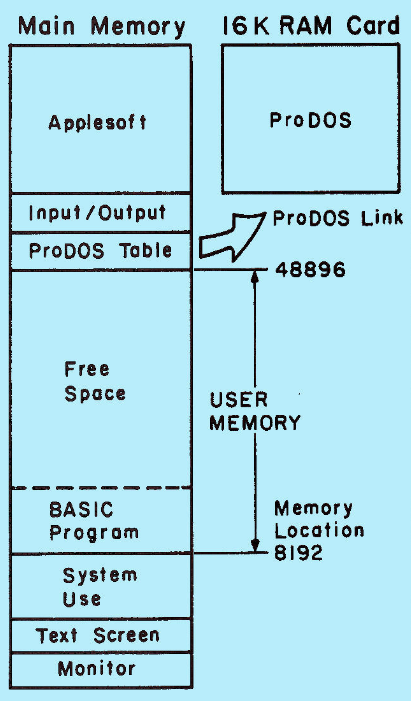


Table 1. *Using selected ProDOS commands in a BASIC program.*

Command	**Use**
RESTORE	If you've created variables in another BASIC program and saved them using ProDOS's STORE command, you should recall those variables early in the program with the RESTORE command.
HIMEM	ProDOS has a moving HIMEM (described in more detail in **Table 2**). If you set HIMEM in your program, you must do so before you open any files and before you declare any strings. If you want to use HIMEM to protect a machine-language program from being overwritten by BASIC, see Appendix A of the *ProDOS Technical Reference Manual*.
PREFIX	In ProDOS, you have to use the full path name every time you access a file. If you're going to access only a few volumes or subdirectories, you should set the prefix to the volume and subdirectory containing the files you'll be using.
/RAM	If you have a 128K Apple (Apple //c or //e with an extended 80-column card), ProDOS lets you use the extra memory as a "RAM disk": The computer loads your data files into the extra memory and accesses the data from memory rather than from disk. Treat the RAM disk as you would any other disk drive. At the start of the program, you'll have to read text files from your floppy disk and write them to /RAM; but, once you've done this, access will be much faster. Before you end the program, you must write that data back to the floppy disk. Otherwise, you'll lose it.
CLOSE	ProDOS doesn't automatically close files at the end of a BASIC program as DOS 3.3 does, so you should insert a global close (PRINT D$;"CLOSE") into your program before you end it.

Table 2. *Applesoft commands that react differently under ProDOS and DOS 3.3.*

Command(s)	**Operation**
HIMEM	The HIMEM command sets the upper limit on the memory BASIC has available for programs. In DOS 3.3, the location of HIMEM is permanent (unless you change MAXFILES). In ProDOS, the location is flexible to accommodate files as they are opened and closed. This may cause problems for machine-language programs written for DOS 3.3 but run in ProDOS. It won't cause problems for BASIC programs, though, because ProDOS moves variables around in memory to accommodate the moving HIMEM.
HGR, HGR2, and TEXT	ProDOS reserves the graphics-screen memory areas for graphics. HGR and HGR2 start this reservation, TEXT cancels it.
INPUT	Applesoft's INPUT statement doesn't allow commas or colons in the string you are entering. Complex data-entry routines using GET are required. ProDOS corrects this deficiency.
IN# and PR#	The IN# and PR# commands switch control of the input and output, respectively, for each of the eight peripheral slots (seven plus the keyboard/video screen) in an Apple II. The line: `999 PRINT D$;"PR#1"` activates the interface card (usually a printer interface card) located in slot 1.
TRACE and NOTRACE	The Applesoft TRACE command lets you watch a BASIC program execute step by step. In DOS 3.3, however, when TRACE is on, DOS 3.3 commands are not executed. With the ProDOS TRACE command, ProDOS commands are executed. NOTRACE still turns the TRACE function off.
FRE	ProDOS's FRE command is superior to Applesoft's. If you put the FRE command in a PRINT statement with a control-D, the computer will pass the command to ProDOS rather than to Applesoft: `10 PRINT CHR$(4);"FRE"`

● Programs that use strings extensively. The ProDOS FRE command concatenates (cleans up) old string garbage much faster than the Applesoft BASIC FRE command, which DOS 3.3 uses.

● Programs that use large text files or are constantly reading information from disk or writing it to disk. Not only do you benefit from ProDOS's faster disk access, but with ProDOS you can use a hard disk.

● Programs that can run on an Apple ///. ProDOS and SOS (the Apple ///'s operating system) have compatible text-file formats. (Although their text files are compatible, the BASIC programs are not.)

Conversely, you shouldn't try to convert other programs:

● Programs written in Integer BASIC. ProDOS doesn't support this language.

● Programs that use the INIT command to format disks from within the program. In ProDOS, you must format blank disks with the Format utility on the master disk.

You can run simple Applesoft BASIC programs on ProDOS merely by converting them from DOS 3.3 to ProDOS. (See Part 2 of this series or pages 103–119 of the *ProDOS User's Manual.*) But ProDOS is different enough from DOS 3.3 that some programs present problems. You may have to change some of the commands in your Applesoft BASIC program to make it run in ProDOS. If programs are difficult to convert, you may want to leave them in DOS 3.3.

The easiest way to find out if an unmodified program will run in ProDOS is to try it. Run a few tests to make certain the program isn't producing garbage as output. If it doesn't run, you'll have to change all incompatible statements.

You should now have a good understanding of ProDOS commands and how to use ProDOS in BASIC. Next month, I'll wrap up this study of ProDOS with an examination of the differences between BASIC programs in DOS 3.3 and ProDOS, including a discussion of all the new errors ProDOS will let you make. ■

Lee Swoboda can't answer all your questions individually. If the answer isn't lengthy and you include a stamped, self-addressed envelope, he'll do his best to respond. Answers to other questions will appear in future articles or in inCider's *Letters column. You can write to Lee c/o Padapple Computer Consulting, 1451 NE Paulson Road, Poulsbo, WA 98370.*

Using ProDOS Part 6

Operating in Harmony

by Lee Swoboda

Now that you're familiar with ProDOS directory structure and the use of ProDOS commands in BASIC programs, let's examine ProDOS' compatibility with existing programs and some of the new errors you may run into. Although most Applesoft commands work the same way in ProDOS as they do in DOS 3.3, you will have to alter others (see **Table 1**). A discussion of some of these changes follows:

● **Control-D.** Because of a curious interaction between the Applesoft GET command and DOS 3.3, some programs insert a carriage return, CHR$(13), before control-D, CHR$(4). Use only control-D in ProDOS.

● **File-access commands.** In ProDOS, you can use semicolons within a PRINT statement to join parts of it, but you can't end the statement in a semicolon.

● **Volume parameter.** With DOS 3.3, you can initialize a disk with a specific volume number from zero to 254, and reference it with a V parameter in your file-access commands. The volume name in ProDOS completely negates the need for a volume parameter.

Switching from DOS 3.3 to ProDOS means modifying your programs.

● **Chaining.** The DOS 3.3 System Master disk contains a binary program called Chain. Using it is rather involved (see p. 106 of *The DOS Manual*). The ProDOS CHAIN command is a simplified simulation of this process.

● **Ending a program.** The END statement is optional in both DOS 3.3 and ProDOS. ProDOS, however, doesn't automatically close text files when the program ends—you must insert a CLOSE statement at each exit.

● **Formatting.** ProDOS has no single command like DOS 3.3's INIT to format a blank disk (see Part 2 of this series). You will have to rewrite any part of a BASIC program that uses the INIT command.

● **File buffers.** DOS 3.3's MAXFILES command lets you vary the maximum number of files that may be open at one time from one to 16. ProDOS allows a maximum of eight open files, with no provision to vary that number.

● **Garbage collection.** ProDOS' special FRE command clears old strings from memory. (See Part 4 of this series for further information.)

● **Free memory.** As in DOS 3.3, the ProDOS formula also lets you determine the amount of random access memory left after the computer allocates space to your BASIC program and variables.

New Errors

With ProDOS, as with DOS 3.3, some errors are related to the operat-

Table 1. Conversion of DOS 3.3 Applesoft programs to ProDOS.

Subject	DOS 3.3 Format	ProDOS Format
*Control-D	D$ = CHR$(13) + CHR$(4)	D$ = CHR$(4)
*File access	PRINT D$;"OPEN"; PRINT F$	PRINT D$;"OPEN";F$
*Volume name	V0–V254	/VOLUME.NAME
*Chaining	PRINT D$;"BLOAD CHAIN, A520":CALL 520;"FILE 2"	PRINT D$;"FILE.2"
*Ending	END	PRINT D$;"CLOSE":END
	PRINT D$;"FP"	PRINT D$;"CLOSE":NEW:END
*Formatting	PRINT D$;"INIT. . ."	Not allowed
*File buffers	PRINT D$;"MAXFILES*n*"	Ignored
Monitor DOS	PRINT D$;"MON C,I,O"	Not allowed
	PRINT D$;"NOMON C,I,O"	Ignored
I/O slots	IN#*slot*	PRINT D$;"IN#*slot*"
	PR#*slot*	PRINT D$;"PR#*slot*"
*Garbage	X = FRE(0)	PRINT D$;"FRE"
*Free memory	PRINT FRE(0)	PRINT D$;"FRE":PRINT PEEK (111) – PEEK(109) + 256* (PEEK (112) – PEEK (110))
HIMEM		Use only when no files are open and no strings declared
PEEK, POKE, or CALL		Check the address being referenced carefully. ProDOS addresses are different from DOS 3.3
BLOAD or BRUN		Check the address and length to avoid conflict with ProDOS or HIMEM locations
Path names	PRINT D$;"OPEN *file name*"	PRINT D$;"OPEN *path name*"
Catalog	PRINT D$;"CATALOG"	PRINT D$;"CAT"

*Discussed in text.

ing system and others to the language (Applesoft BASIC). The new ProDOS commands introduce additional possibilities for error. The error-handling procedure for ProDOS, however, is similar to that for DOS 3.3: The computer beeps and prints an error message on the screen. **Table 2** summarizes the errors you can make with each ProDOS command, and **Table 3** explains their probable causes. Below is a more detailed explanation of the errors you might encounter when using ProDOS's new commands:

● **No device connected.** This error occurs if you use the slot/drive parameters in a ProDOS command with no interface card in the slot, or if you specify a peripheral slot number in the IN# or PR# command when the peripheral device isn't connected.

● **Path not found.** If the computer can't find the path name you specify, check your spelling and make sure the file actually exists. Also, check that you didn't use a partial path name that is invalid for the current prefix, or that you didn't change floppy disks, making the current prefix invalid.

● **Invalid option.** Make sure you use the correct options with ProDOS commands (see **Tables 4** and **5**).

● **Directory full.** You can't add (SAVE, BSAVE, CREATE, or STORE) more than 51 files to the volume directory. If you have additional files, you should rearrange your disk to place more files in subdirectories, since ProDOS doesn't limit their size.

● **File not open.** You must first open a file with the OPEN command before you can perform an operation (POSITION, READ, or WRITE) on it. The DOS 3.3 READ and WRITE commands automatically open and close files before performing the operation.

● **Duplicate file name.** You can't create or rename a file with a name that already exists.

● **File busy.** It is difficult to destroy data in files with ProDOS; for instance, it won't let you perform an operation (CAT, CATALOG, DELETE, or RENAME) on a file that is still open. You must open files yourself, and close them later.

● **File(s) still open.** This message occurs when an error or control-C halts program execution without closing open files. You must close open files from the immediate mode before you can load or run the next program.

Benchmarks

Since ProDOS accesses files and collects garbage faster than DOS 3.3, programs that use strings and disk access extensively usually execute faster under ProDOS. Results vary with the circumstances, but from **Table 6** you can see that ProDOS is consistently faster than DOS 3.3 in all file-access operations, especially in accessing text files. That additional speed, along with the new INPUT and FRE commands, brings Apple up to par with CP/M- and MS-DOS-based computers in ease of handling.

Where Do You Go from Here?

In this series, I've condensed the most important aspects of ProDOS, but I've only scratched the surface. Here are some additional topics you might explore:

- **Machine-language interface (MLI).** Machine-language routines for assembly-language programmers are easy to access with ProDOS.
- **File structure.** You can use a hard disk on your Apple II to avoid worrying about the size of your files.
- **Treatment of text.** ProDOS handles text differently from DOS 3.3.
- **Interrupts.** ProDOS can handle interrupts—signals from another device that it needs attention; DOS 3.3 can't.
- **Directory structure.** The internal structure of ProDOS's directories is different from that of DOS 3.3's.
- **Disk arrangement.** The ProDOS disk, like the DOS 3.3 disk, has 35

Table 2. *ProDOS commands and error messages.*

	Error Message																				
Command	Range error	No device connected**	Write protected	End of data	Path not found**	I/O error	Disk full	File locked	Invalid option**	No buffers available	File type mismatch	Program too large	Not direct command	Syntax error	Directory full**	File not open**	Duplicate file name**	File busy**	File(s) still open**	Undef'd statement error	Bad branch error
APPEND	●	●	●		●	●		●	●	●	●		●	●							
BLOAD	●	●			●	●			●		●	●		●							
BRUN	●	●			●	●			●		●			●							
BSAVE	●	●	●		●	●	●	●	●					●	●						
*CAT	●	●			●	●			●	●	●			●				●			
CATALOG	●	●			●	●			●	●	●			●				●			
*CHAIN	●	●			●	●			●		●	●		●						●	●
CLOSE						●			●					●							
*CREATE	●	●	●		●	●	●		●					●	●		●				
DELETE	●	●	●		●	●		●	●					●				●			
EXEC	●	●			●	●			●	●	●			●							
*FLUSH	●					●			●					●							
FP														●					●		
*FRE														●							
INT														●					●		
IN#	●	●							●					●							
LOAD	●	●			●	●			●		●	●		●							
LOCK	●	●	●		●	●			●					●							
OPEN	●	●	●		●	●	●		●	●	●		●	●	●						
POSITION	●			●		●			●				●	●		●					
*PREFIX	●	●			●	●			●		●			●							
PR#	●	●							●					●							
READ	●			●		●			●				●	●		●					
RENAME	●	●	●		●	●		●	●					●			●	●			
*RESTORE	●	●			●	●			●		●	●		●							
RUN	●	●			●	●			●		●	●		●						●	●
SAVE	●	●	●		●	●	●	●	●		●			●	●						
*STORE	●	●	●		●	●	●	●	●		●			●	●						
UNLOCK	●	●	●		●	●			●					●							
WRITE	●		●			●	●	●	●				●	●		●					
*DASH(-)	●	●			●	●			●	●	●	●		●				●	●	●	

*New commands

**New ProDOS errors

tracks with 16 sectors per track, but ProDOS uses them differently.

Additionally, Apple publishes three excellent manuals on ProDOS, each with an accompanying disk:

- *ProDOS User's Kit* (Apple product A2D2010, $40), including the *ProDOS User's Manual*, *ProDOS Supplement to Apple //e Owner's Manual*, and ProDOS User's Disk (the ProDOS master disk). The manual covers the use of ProDOS utilities.
- *ProDOS BASIC Programmer's Kit* (Apple product A202037, $35), including the *BASIC Programming with ProDOS* manual and ProDOS BASIC Programming Examples disk. The manual covers the use of ProDOS in BASIC. The disk provides examples of BASIC programs that use ProDOS commands, and also includes the Applesoft Programmer's Assistant (APA), which lets you renumber and merge Applesoft programs.
- *ProDOS Technical Reference Workshop* (Apple product A2W0010, $25), including the *ProDOS Technical Reference Manual* and ProDOS Exerciser disk. The manual thoroughly covers the technical details of ProDOS—the location of all bytes. The disk leads you through step-by-step instructions for using ProDOS from machine language.

These six articles should help those of you who own ProDOS to use it more effectively and understand how it differs from DOS 3.3. ProDOS is a sophisticated disk operating system, and is a significant improvement over DOS 3.3—I hope this series will influence those of you who don't already own ProDOS to buy it.■

Lee Swoboda can't answer all questions individually. However, if the answer is not lengthy and you include a stamped, self-addressed envelope, he'll do his best to respond. Lee Swoboda's address is Padapple Computer Consulting, 1451 N.E. Paulson Road, Poulsbo, WA 98370.

Table 3. *ProDOS errors and probable causes.*

Error Message	Code**	Applesoft	ProDOS	DOS 3.3	Probable Cause
Bad subscript	107	•			Array subscript beyond DIM'd value
*Directory full	17		•		Attempt to add more than 51 files to volume directory
Disk full	9		•	•	No more storage space on disk
Division by zero	133	•			Divisor in Applesoft formula is zero
*Duplicate file name	19		•		Attempt to CREATE or RENAME file that already exists
End of data	5		•	•	Attempt to READ beyond last record of TXT file
*File busy	20		•		Attempt to OPEN file that is already open
File locked	10		•	•	Attempt to WRITE to locked file
*File not open	18		•		Attempt to READ or WRITE closed file
File type mismatch	13		•	•	Attempt to use invalid command for file type
*Files still open	21		•		Last program ended without closing files
Formula too complex	191	•			More than two IF. . .THEN conditions executed at once
I/O error	8		•	•	Disk drive door open, no disk in drive, or disk unformatted
Illegal quantity	53	•			Numeric value beyond allowable range
*Invalid option	11		•	•	Invalid optional command parameter
Language not available	1			•	Attempt to run integer BASIC program
NEXT without FOR	0		•	•	More NEXTs than FORs
No buffers available	12		•	•	Attempt to OPEN more than eight files
*No device connected	3		•	•	Attempt to access an empty peripheral slot
Not direct command	15		•	•	Command must be used in deferred mode
Out of data	42	•			Attempt to READ past last DATA statement
Out of memory	77	•			Program too large or too many variables
Overflow	69	•			Attempt to use number greater than ± 1.7E38
*Path not found	6		•	•	No path with indicated name
*Path not found	7		•	•	No path with indicated name
Program too large	14		•	•	Insufficient memory for CHAIN
Range error	2		•	•	Optional command parameter beyond allowable value
REDIM'd array	120	•			Attempt to use DIM statement for previously DIM'd array
RETURN without GOSUB	22	•			More RETURNs than GOSUBs
String too long	176	•			Attempt to build string longer than 255 characters
Syntax error	16	•	•	•	Spelling or punctuation error
Type mismatch	163	•			String and numeric variables mixed in same operation
Undef'd function	224	•			Used FN statement without DEF FN
Undef'd statement	90	•			Attempt to GOTO nonexistent line number
No message displayed	254	•			Invalid response to INPUT statement
No message displayed	255	•			Control-C program interruption

* New ProDOS error

** Value obtained using Applesoft statement "PRINT PEEK(222)"

Table 4. *Valid options for ProDOS commands.*

ProDOS Command Format

BLOAD path name,*An,Bn,Ln,Txxx,Sn,Dn*	IN#n
BLOAD path name,*An,Bn,En,Txxx,Sn,Dn*	IN#An
BRUN path name,*An,Bn,Ln,Sn,Dn*	LOAD path name,*Sn,Dn*
BRUN path name,*An,Bn,En,Sn,Dn*	LOCK path name,*Sn,Dn*
BSAVE path name,An,Ln,*Bn,Txxx,Sn,Dn*	POSITION path name,Fn
BSAVE path name,An,En,*Bn,Txxx,Sn,Dn*	POSITION path name,Rn
CAT *path name,Sn,Dn*	PREFIX *path name,Sn,Dn*
CATALOG *path name,Sn,Dn*	PR#n
CHAIN path name,*@n,Sn,Dn*	PR#An
CLOSE *path name,Sn,Dn*	PR#n,An
CREATE path name,*Txxx,Sn,Dn*	RENAME path.1,path.2,*Sn,Dn*
DELETE path name,*/Sn,Dn*	RESTORE path name,*Sn,Dn*
EXEC path name,*Fn,Sn,Dn*	RUN path name,*@n,Sn,Dn*
EXEC path name,*Rn,Sn,Dn*	SAVE path name,*Sn,Dn*
FLUSH *path name,Sn,Dn*	STORE path name,*Sn,Dn*
FRE	UNLOCK path name,*Sn,Dn*
	- path name,*Sn,Dn*

Sequential Text-File Commands

APPEND path name,*Txxx,Sn,Dn*	READ path name,*Fn,Bn*
OPEN path name,*Txxx,Sn,Dn*	WRITE path name,*Fn,Bn*

Random Access Text-File Commands

APPEND path name,*Ln,Sn,Dn*	READ path name,Rn,*Fn,Bn*
OPEN path name,*Ln,Sn,Bn*	WRITE path name,Rn,*Fn,Bn*

Roman portions are mandatory; italic portions are optional.

Table 5. *Allowable values for options shown in **Table 4**.*

Description	Syntax	Range of n
Slot number	,Sn	1 to 7
Drive number	,Dn	1 or 2
Field number	,Fn	0 to 65535
Record number	,Rn	0 to 65535*
Number of bytes	,Bn	0 to 65534**
Address in RAM	,An	0 to 65535
Length in bytes	,Ln	1 to 65535
End address in RAM	,En	1 to 65535
At line number	,@n	0 to 65535
Slot number***	n	0 to 7
File type	Txxx	see Part 2

* Rn is limited to 16 megabytes divided by record length (L) or 65535, whichever is less.

** Bn is limited to one less than the record length (L – 1) or 65535, whichever is less.

*** Used in the IN# and PR# commands.

Table 6. *Faster execution of certain commands under ProDOS.*

Command	ProDOS Execution Rate (Times Faster than DOS 3.3)
WRITE	3–4
READ	5–6
APPEND	3–4
SAVE	2
LOAD	5–6
BLOAD	3–4
BSAVE	2