ProSel-16 documentation

Overview:

ProSel-16 consists of four main files:

```
   START, the main program and utilities, a large file presently
               about 217 blocks (224 blocks if "Zipped").
   PROSEL.SPECS, information about application programs that
               appear on the main screen of ProSel-16, variable size,
               at most 16 blocks.
   PROSEL.PARMS, optional file containing user modifiable parameters.
               It is up to 6 blocks long.
   PS.16.TO.8, a file for technical use (you don't have to know
               anything about it) when running P8 programs.  3 blocks.
```

The START file is meant to be in the SYSTEM subdirectory of the boot volume. That is the only way it will come up on booting the system. However, it could be put elsewhere and its name can be changed if you don't want it as the boot program.

The PROSEL.SPECS file should be in the main directory of the boot disk. It, and copies of variations of it, CAN be on the main directory of non-boot disks, but it must be on some disk accessible at boot time.

PROSEL.PARMS and PS.16.TO.8 should be in the same directory holding START, usually the SYSTEM subdirectory of the boot disk. These files are optional - the system will run without them, but they are recommended strongly.

In addition to these files, the system uses two subdirectories (anywhere), one of which holds "transient commands" and the other of which holds help files.

When you boot, the ProSel screen will appear. It holds titles of your application programs, in a very flexible, user definable format. To run a program you merely have to use the cursor keys or the mouse to highlight one of them and then press RETURN or click the mouse. There are several other ways of running a program but you need not be concerned with them at first.

Pressing the ESC key will bring up the "main menu" which has several selectable items. You can go from there to the ProSel-16 utilities (a comprehensive set of file/volume manipulation features that are very powerful yet easy to use). Or you can go to the editor, which is used for changing the main screen, adding items to it, or deleting items from it. There are several other options from the main menu that will be described later.

ProSel-16 requires some memory enhancement, but should run with at least 500k of expansion memory. ProSel-16 runs under GS/OS.

Installation:

If you have a hard disk then you should run the installation program. Just boot the ProSel-16 disk and press G <RTN> to select the item "Greetings & installation". When the installation is finished you should reboot because ProSel-16 uses the boot disk for some things, and, until you reboot, that disk will be the ProSel-16 disk.

The installation gives three options, minimal (ProSel-16 only), partial, or full installation. The latter two install ProSel-8 (if not already installed) as well as ProSel-16, but the partial installation installs only the most used parts of the system. (Even the "full" installation does not install everything and there may be some items you will want to copy over manually.)

The full installation requires almost 495 free blocks on the disk, the partial one requires about 350 blocks, and the minimal one requires about 250 free blocks.

The installation creates the subdirectories UTIL and PROSEL.16 and the subdirectories COMMANDS and HELP.FILES of PROSEL.16. The full installation copies the command files to the COMMANDS directory, but does not copy the help files. On a hard disk, you will want to copy the help files also.

If you do not have a hard disk, then do the installation on a copy of your system disk, but first get rid of any items on the disk that you can do without. Check the room on the disk to see whether there is room for one of the installation options. (The installation also checks for sufficient room and will not attempt to do the installation if enough room is not available or if the destination disk is not a GS/OS boot disk.)

If you want the opening super-hires logo, that appears when you boot the ProSel-16 disk, to appear when you boot your disk, just copy the file InitPic from the SYSTEM/SYSTEM.SETUP directory to that directory of your boot disk. It should not prolong booting by more than a second on a hard disk.

IMPORTANT NOTE: The installation does not install GS/OS. You should already have installed GS/OS on your boot disk. If you have not done so yet, I recommend using the INSTALLER on your GS/OS system disks to do it.

GS/OS almost requires a hard disk for efficient use, or at least a large ROM disk to boot from. If you don't have one of these, you are strongly advised to consider these options.

Getting started:

Let's assume that you have installed ProSel-16 on your boot disk and have rebooted and are now looking at the ProSel screen. Let's also assume that you are new to ProSel. Then the first thing you are likely to want is the set of ProSel help files. Since the installer has not copied those files, let's do that now.

Press the ESC key. You now see a menu, and one of the items is Utilities. You can either move the cursor to that item using the up/down arrow keys or the mouse, or you can simply press U for Utilities. If you used one of the former methods then press RETURN or click the mouse button.

You will note some disk activity, but shortly the Utilities menu will appear. One of the items is X-Copy files. (X is used since C is taken for the Catalog function.) Either highlight that item with the cursor keys or the mouse and press RTN/button, or just press X. The copy files screen will appear with some instructions. Let's just ignore the instructions for now. The cursor indicates a request for a source pathname. Since the help files are in the /PROSEL/PROSEL.16/HELP.FILES directory, type in that name. Then a destination is asked for. The default will probably be your boot disk name, so press TAB which will move the cursor across the volume name (or volume part of the default name). Here you can just type in the name of the directory you want to copy to (which is PROSEL.16/HELP.FILES). But let's assume we have forgotten the name of the directory to which we want to copy the files. Press the ? key (remember to SHIFT) or just click the mouse button, making sure the cursor is just after the volume part of the pathname. The disk will be accessed for a while, and then you will see a "tree diagram" of all the directories on your disk. Use the cursor keys or mouse to move to the desired directory (which is PROSEL.16/HELP.FILES) and then press RETURN or click the mouse.

Next you will see a list of files from the source directory. They are the help files. Press the space bar. You will see that a check mark has been placed on the first file and that the cursor has moved to the next file. (Instead of the space bar, you could have clicked the mouse button to get the same result.) You could do this for all the files, but there is an easier way to select all the files: move the cursor back to the checked file and press space again, removing the check mark. Now press control-A. (This is often denoted by ^A, a notation we shall use from now on. Remember that the control key must be held down while the A key is pressed and released.) On pressing ^A you will see that ALL the files are now checked. (A stands for ALL.) However, no files have yet been copied - this is just the selection phase of copying files. Now press RETURN. You will see all the files being copied. Each one is highlighted just before it is copied. If only some of the files had been selected then only those would be highlighted.

When the copy is done, you will be returned to the Utilities main menu, and you can exit by pressing RTN (the cursor is on Quit) or by pressing the Q key or the mouse button.

You should now be back at the main ProSel screen. The next thing you are going to want to do is to place some of your applications on the ProSel screen. To do that you must use the editor. Since we just copied the help files let's see if there is some help on the editor. If there is an item called Help on the screen, select it by highlighting it and pressing RETURN or clicking the mouse. If not, the hold down the Apple key and press the H key, and then RTN. (From now on, I will indicate this by "press OA-H <RTN>.) This will cause some disk activity and then a list of help files will appear. To select the one called EDITOR just type ED <RTN>. (The help manager only requires a few letters of the help file name to understand which one you want, so it is not necessary to type out the complete name EDITOR.) This will display the help information on the editor. Quite frankly this information will probably not make much sense to you at this time, but you may find it useful after you get more used to using ProSel.

Let's go to the editor. If there is an item ProSel-16 editor on the screen, select it. If not, you can still go to the editor by pressing OA-E <RTN>, or by pressing ESC and selecting Editor from the main menu.

The editor will come up after some disk activity and will display a screen much like the ProSel screen itself, except that the text "ProSel editor" will be seen at the bottom of the screen.

You are going to want to add an application to the screen, so move the cursor to an empty spot on the screen and press RETURN. (Sorry, the mouse is not supported in the editor at this time.)

This will bring up a screen with a great deal of text. Read it if you like. It also presents you with two choices (A)utomatic or (M)anual mode. The manual mode assumes that you already have some knowledge of pathnames and how ProSel works. Also, you have to know exactly where your desired application is on the disk. OK, so let's try Automatic mode instead: press A.

It now asks for a directory to search and will have the volume name as a default. Giving a directory name will limit the search and also make it faster, but you can just press RETURN at this point to have it search the entire disk. Do that. The disk will run for a while and then a list of files will be shown on the screen. These are all the executable files on the volume, or if the screen is full it is all of them that will fit. (You could have used a subdirectory instead to get names that may not fit when you use the volume name.)

Look for a file that appears to be the program you want to add to the ProSel screen. Use the cursor keys to highlight it and press RETURN. That is it, it is done! You will now be looking at the main editor screen and the name of the application will now be there under the cursor. Do you want to change the name to something prettier? If so, press RETURN. What you see now is manual mode, but the specifications for the application will all be there. The cursor will be on the "screen title" and you can change that to whatever you want. Do not change the other three items, but ponder them if you want. Later, you will want to understand what they are all about.

Now let's save our changes. Press ESC to return to the main editor screen and another ESC to get to the editor menu. There are two save items, one that is (S)ave as a screen and the other is (P) for save a new PROSEL.SPECS file. It is the P choice that you want. So press the P key.

The save will send you back to the ProSel screen from the editor. You should see the item that you added to the screen. Move the cursor to highlight it. Then press RETURN or click the mouse to run that program.

When you Quit or Exit (or whatever that program calls it) from your application, you should return, after a little disk activity, to the ProSel screen.

That is about all you have to know for the basic operation of ProSel. There is a lot more to ProSel than what we have described, and the rest of this documentation will go into everything. Some of what you will read may be too technical for you at this time. Don't worry about it. When you climb Mt. Whitney, you must first drive to Lone Pine, and then you have to get to Whitney Portal. Both of these are much easier than the trip to the top, and most of us will never reach the pinnacle. The higher we get the harder it is, but, however far we go, the view is magnificent.

The structure of the ProSel-16 system:

We have already met several parts of the ProSel system; the main screen (which you can hardly miss), the main menu, the editor, the Utilities (which corresponds, incidently, to the ProSel-8 program called Cat.Doctor), and the Help facility.

Other main parts of the system are the File Finder, Information Desk, Backup/Restore, the Optimizer, the block editor, and the Shell command processor. The latter is one of those things which present you with a blank screen and a cursor and dare you to do something intelligent. This frightens some people and if you are one of those, let me assure you now that this is a part of ProSel that you don't even have to see if you don't want to. It is there for people who want it. If you land there by mistake and you start shaking, remember the ESC key! Think of that key up in the left upper corner as your panic button - the cure of all evil. Nevertheless, we will have to talk about the Shell later on.
It is one, together with the main screen and the main menu, of the three major "hinges" of the ProSel system. The other parts, such as the Utilities, as important to you as they may be, are more like separate programs that are just a little more immediately accessible than most applications. Indeed, although all these things are in the same file START, only the main screen, the main menu, and the command processor are loaded into memory upon boot. The other parts are loaded whenever you go to them. You noted the fairly sizable disk access when you went to the Utilities. That was loading the Utilities (a sizable hunk) from disk into the computer. After you quit from Utilities back to the main screen, if you again went to Utilities, you would go there very quickly, without all that disk access. This is just from the fact that Utilities has already been loaded by the previous access, and now does not have to be loaded again.

Here is a diagram that may help you understand the relationships between parts of the program. It does not show everything; ProSel, being a very flexible program, usually has more than one way of accomplishing a particular goal.

```
   /------------> External applications <------------\
   |              ---------------------              |
   |                                                 |
   |Select                                           |Command
   |                                                 |
   ^            ESC                  S               ^
MAIN SCREEN <---------> MAIN MENU >-------> COMMAND PROCESSOR (SHELL)
   v    ^                   v                   v    v
   |    |___________________|___________________|    |
    \                       |       ESC             /
     \                      |                      /
      \                     |                     /
       \OA-letter <RTN>     |Selection           /Command
```

```
        \                   |                   /
         \                  |                  /
          \                 |                 /
           \                |                /
            \               |               /
             \              |              /
              \             |             /
               \            |            /
     ^          \           |           /          ^
     |           \          |          /           |
     |            \         |         /            |
     |Quit         \        |        /             |Quit (depends on
     |              |       |       |              |  access method)
     |              v       v       v              |
     `-----------< Subsidiary modules >------------'
                           ------------------
                                Utilities
                                Editor
                                File finder
                                Info desk
                                Backup/Restore
                                Optimizer
                                Zap
                                Etc.
```

On the other hand, this may just look like a lot of hot air to you. If so, never mind, perhaps it will be clearer later.

Executable files:

ProDOS files can have 256 different "filetypes" (or simply "types"). The more common filetypes have names (always three characters). Some types are reserved and have no presently defined file structure, and these are indicated on catalogs by a hex byte such as $F9. Presently, programs that can be run directly can have one of three types, called:

SYS (system file)
S16 (system-16 file)
EXE (shell executable file)

The first of these, SYS is the filetype of executable files that run under ProDOS-8. Its counterpart under ProDOS-16 or GS/OS is a S16 file. EXE files are also GS/OS executable files, but often expect a "shell", such as ProSel-16, to be in memory for various types of assistance. The EXE files run "under" a controlling environment, know as a shell. Unfortunately, programmers do not always follow the rules for these files, and so there are many SYS files around that are not really stand-alone executable files. Also there are many EXE files that do not behave according to the design of the EXE filetype.

These three types are the filetypes of files that can be run from ProSel. In addition, there are BAS (BASIC) files which are designed to run under BASIC.SYSTEM. In this case BASIC.SYSTEM is the SYS file that is run, and the BAS file's name is passed by the selector to BASIC.SYSTEM when the latter is run, telling it to run, in turn, the BAS program. Similarly BIN (BINARY) files often, but not always, are programs designed to run under BASIC.SYSTEM and they are handled in the same way as BAS files by ProSel. Rarely, TXT (TEXT) files are designed to run under BASIC.SYSTEM as "exec" files. Since this is so rarely the case, and since such files are easily run from a simple BASIC program, I have decided not to allow direct access to these from ProSel-16. Instead, ProSel-16 will EXEC such a file as a set of shell commands under PROSEL rather than BASIC.SYSTEM. [See the section on the Command line processor, and particularly on "automatic operation".]

Program selection from the main screen:

It probably does not have to be said that to run a program from the main screen you use the cursor keys or mouse to highlight the desired item and press RETURN or click the mouse button. Also, as indicated on the screen, you can press ESC to go to the main menu. However, there are other actions you can take that are not so obvious.

If you press an alpha key (A-Z) at the main screen, then the highlighting will move to the next item starting with that letter. The case of the letter has no effect. If there is no item beginning with that letter then the cursor moves to the first item on the screen. Scanning is done downwards and then across.

If you press a number key on the main keyboard then the disk DEVICE corresponding to that number (0 is used to mean 10 for this purpose) will be accessed and a list of the executable files and subdirectories from that disk device will be shown on the screen. You can select an executable file to run, or you can select a subdirectory, causing that directory to be read and its files displayed, etc.

> DEVICE NUMBERS: These are the GS/OS or P16 equivalents of the P8 slot/drive. Device #1 is always your boot disk. The others depend on how your SYSTEM/DRIVERS directory is set up. Some devices are not disk drives at all - for example the CONSOLE (screen) may be a device. To see the correspondence between devices and their numbers, use the P command from the main menu or OA-P <RTN> from the screen.

If you press a number key on the KEYPAD, then the directory (if any) corresponding to that PREFIX will be read, and its executable files and subdirectories will be displayed for selection as above.

> PREFIXES: GS/OS keeps a set of 33 "prefixes", which are names of some special directories. They are named */, 0/, 1/,...,31/. The */ prefix is always the boot volume name, The 0/ prefix is the "default" prefix, meaning that it is added to any pathname that is not a "full" pathname. The 1/ prefix is the directory containing the current application file. When in ProSel-16 it is */SYSTEM if ProSel-16 is installed in that directory as recommended. Prefix 6/ is used by ProSel-16 as the "transient command" directory, and prefix 7/ is used by ProSel-16 as the "help files" directory. All these prefixes, except *, can be set up as desired by the user by modification of the ProSel parameters, as will be detailed later.

The option key together with a number key adds 10 to the key value for both of these provisions. Similarly the Apple key adds 20 and control adds 40 (main keys only) with combinations accumulating.

In addition, if you hold the Apple key down when typing an alpha key, you will go to the main menu with the cursor on the item corresponding key. (If none, no action occurs.) For example, OA-U goes to the main menu and highlights the Utilities item. Thus pressing RETURN then sends you to the Utilities module.

When you run a program from the main screen, there are also some options that can be requested by pressing one or more of the Apple key, Option key, Shift key, and Control key. (These can also be made automatically selected options via the program "specs", to be described later.)

(1) Apple key down. This will cause ProSel-16 to do a memory purge before running the program. This can be useful for running some very large programs that may not be able to get enough memory after the loader has already loaded the program.

(2) Option key down. This will cause ProSel-16 to purge memory and to force reloading of itself after the application has quit. This has the effect of cleaning up memory a little more thoroughly than just case (1). It MAY allow running programs for which (1) is insufficient.

(3) Apple and Option keys down. This will do the same as (2) but will also not request return when the application quits. This would be appropriate if you wanted it to return to another selector.

(4) Shift key down. This has effect only when running a SYS program, that is, a P8 application. This will cause ProSel-16 to set up

ProSel-8 before launching the program, so that it will be ProSel-8 which gains control when the application quits. This would be appropriate if you want to run more than one P8 program since it would prevent the time consuming passage from P8 to P16 and back. Of course, this requires ProSel-8 to be available. In fact it supposes that the PROSEL.SYSTEM and the PROSEL files (both part of ProSel-8) be in the main directory of the boot volume. [To do this automatically from a program's specs (see below) you would set the specs to run PROSEL.SYSTEM as the application and the desired SYS application as the "startup".]

NOTE: If there is a startup specification in the program's specs then the shift key will be ignored because there is a conflict between these two things. However, holding down the control key defeats the startup, so both control and shift down will ALWAYS force a shift to ProSel-8.

(5) Control key down. This overrides any "startup" in ProSel-16's specification list for the file. Thus, if you have an application specification that instructs the application to automatically open a certain file, then running that program with the control key down will bypass the open instruction. You only have to hold the control key down for a moment while you press RETURN or click the mouse. You can let it up as soon as the "launching" message appears.

Line input:

At various places in ProSel-16 you must type a line specifying a directory or other information. This occurs, for example in Utilities, the Editor, and the Command processor. In all cases line input supports a number of line editing features:

```
^D     deletes character under the cursor.
DELETE deletes the character to the left of the cursor.

^E     toggles insert mode.
OA-key inserts character whether or not in insert mode.

TAB    moves just past next / , or goes to line end adding /.
OA-TAB moves back to right of previous / or to line start.
^N     moves to line end.
^B     moves to line beginning.

^Y     clears line from cursor forward.
CLEAR  clears line cancelling all previous input.

ESC    aborts.
```

When you are asked for a device number (which happens in some parts of Utilities, Find file, and Info desk), you can press the ? key. This will cause a list of up to 10 recognized device names to appear and you can use the cursor keys to select one of those. In that mode, if you press the ? key at any time, the name of the volume occupying the highlighted disk device will be displayed. For devices not on the list you can input the device NAME, starting with a period, at the device number prompt. An example of a device name is ".APPLEDISK3.5A". You can also input a two digit device number after a period at any device number prompt. You can also input the VOLUME name of the volume in the device, starting with /.

Some parts of ProSel support additional features for line input. For example, Utilities uses ^X to switch between the two default pathnames. (Elsewhere ^X is the same as CLEAR.) Similarly, in the command line processor, the up/down arrows take you through the history of commands, while, in other places, these keys are not accepted and do nothing.

The main menu:

The main menu is accessed from the screen by pressing ESC. It contains a number of options:

```
Help - brings up a list of help topics from which you can select.
Quit - returns to a program that ran ProSel-16, if any.  If none,
                then it just reruns ProSel-16 (under GS/OS) or puts up
                an Apple designed option screen (under P16).
Editor - enters the ProSel-16 application list editor.
Utilities - enters the file utilities.
File finder - enters the file finder module.
Show prefixes - try it.  It also shows some other data.
Warm shutdown - use to reboot, or before shutting down system.
Cold shutdown - ditto, but this one puts control panel changes
                into effect if you press return to reboot.
Shell - enters the "shell" or command line processor.
Refresh screen - rereads the program specs file, can be used
                to shift to a specs file on another disk.
Backup/Restore - incremental hard disk backup.
Volume repair - this corresponds to "Mr.Fixit".
Optimizer - ProDOS volume optimizer (Beach Comber).
Exchange screens - this is the "cycler".
Info desk - enters the information desk module.
Modify parameters - allows modification of the user definable
                parameters for ProSel-16 (the PROSEL.PARMS file).
Zap - a block editor called Block Warden.
```

In addition the ESC key returns you to the main screen.

You have already met some of these options. Most of them will be described in more detail in later sections.

As mentioned before, pressing OA-x while at the main screen and where x is the command key for one of these menu options, will take you directly to the menu with the cursor on the desired item. Although this is not less keystrokes than the regular ESC and then selection of the item, it will be more intuitive to some people.

When you choose any of the options on the main menu, and return, you are never sent back to the menu. Instead you go directly back to the main screen.

Also, many of these items can be selected in other ways. They can be set up as selections on the main screen by the proper choice of "specs", as we will discuss later. Some of them can also be accessed directly for the command processor.

Pathnames:

This section is a short tutorial on "pathnames" in ProDOS. It does not go into some of the esoteric additions that have been added to GS/OS.

A pathname is a roadmap telling ProDOS how to find a file on the disk. ProDOS volumes are organized in a "tree" structure, sometimes called a hierarchic structure. The "root" of the tree is the volume name. Off this root are limbs, most of which are subdirectories or "folders". Some of the limbs are ordinary files. The subdirectory limbs can have limbs growing off of them, etc, etc. An example of such a structure is:

```
/VOLUME.NAME/ ----->SYSTEM/ -----------> SYSTEM.SETUP/ ---> TOOL.SETUP
                        |                   |                   '-> TS2
                        |                   |
                        |                   |--> DESK.ACCS/ ------>
CLOCK.CDA
                        |                   |
                        |                   |--> TOOLS/ ----------> TOOL.
001
                        |                   |
                        |                   |--> GS.OS
                        |                   |--> P8
                        |                   |--> START
                        |                   '--> PROSEL.PARMS
                        |
                        |--> APPLEWORKS/ ------> APLWORKS.SYSTEM
                        |                   |--> SEG0
                        |                   |--> SEG1
                        |                   '--> MYDATA/ ---------> MYFILE1
                        |                                       '-> MYFILE2
                        |
                        '--> PROSEL.16/ -------> COMMANDS/ -------> ASCII
                                            |                   |-> WHERE
                                            |                   '-> TYPE
                                            |
                                            '--> HELP.FILES/ -----> DEVICES
                                                                '-> HISTORY
```

Here, you may "climb" the tree by going either right or down (I guess this is a fallen tree) at any crook. In the diagram the titles ending in / are directories (/VOLUME.NAME/ is the root, or main, or volume directory, and the rest are subdirectories). Using the slash to end the name of a subdirectory is just a convenience here, it is usually optional when specifying a directory name.

The other files shown in the diagram are regular (non-directory) files. Some of them, like START, are executable programs, while others like HISTORY are data files. To construct a file's "full pathname", you just climb the tree adding each name to the last. Thus the full pathname of the file DEVICES is

/VOLUME.NAME/PROSEL.16/HELP.FILES/DEVICES

A prefix is a full pathname to a DIRECTORY file. Thus the pathname above is not a prefix since it points to a non-directory, but

/VOLUME.NAME/PROSEL.16/HELP.FILES/

and

/VOLUME.NAME/PROSEL.16/

and

/VOLUME.NAME/

are all valid prefixes. One can SET the system's prefix by various means and then it is called THE prefix, or, in GS/OS, it is prefix #0.

A "partial pathname" is a pathname NOT beginning with a / (or some other characters - see below). Giving a partial pathname for a file means that adding the given partial pathname to the prefix would result in the FULL pathname for the file.

[Aside: When specifying a prefix, one usually does not demand that the / mark at the right hand end be given. It is understood. So, when forming the FULL pathname, one adds the separator /

between the prefix and the partial pathname if it is not already there.]

Thus, if the prefix is set to /VOLUME.NAME/PROSEL.16/ then a valid partial pathname for the DEVICES file is HELP.FILES/DEVICES.

GS/OS supports some additional conventions that make the task of giving a file's pathname easier. As previously mentioned, there are 32 other prefixes that GS/OS remembers, named */, 1/, 2/, ... 31/. The default prefix can also be called 0/. Note that these DO NOT start with the / character.

The prefix */ cannot be changed, and is always the name of the boot volume. In the example, if the tree given is the tree for the boot volume then */ and /VOLUME.NAME/ mean the same thing.

The prefix 1/ is set to the directory containing the currently running program. (It can be changed by the program, but this is not advisable.) Thus, in ProSel-16 it will ordinarily be the same as */SYSTEM/. If you run APLWORKS.SYSTEM in the above example, then the prefix 1/ (or "prefix #1") will be */APPLEWORKS/ which is the same as /VOLUME.NAME/APPLEWORKS/.

Application specifications & the editor:

We have already discussed using the editor to add applications to the screen display by using the "automatic" mode. For most uses this is sufficient, and if "pathnames" confuse you, you you can stick with the automatic mode for the time being and not worry about the more technical details in this section. Here we describe the "manual" mode of the editor, and how to access some of the special options that one has in running applications from ProSel-16.

An application specification in ProSel consists of four items:

(1) The screen title. This is what you see on the screen. It can be anything at all as long as it starts with an alpha character. It can contain control characters to do such things as turn on/off mousetext or inverse. Even the beginning alpha character can be made invisible by following it with a backspace and another character.

If the "screen title" starts with a character whose ascii code is less than $40 (eg., numbers, control characters, and most punctuation characters), then it will not be a selectable item. This can be used for topic headers on the screen, or similar things. The cursor will not land on such an item when the cursor keys are used. The mouse can land on them, but cannot select them, and the mouse button will cause the cursor to move to the next selectable item.

(2) The prefix. This sets prefix #0 (the "default" prefix) for the application. It is usually, but not always, the directory containing the application program. But it could be a subdirectory containing the application's data files, or whatever you want. It could even be a directory on another volume. Note that some applications set this prefix themselves and, in that case, this item does not accomplish anything. However, it is still required and must be a full pathname, such as */APPLEWORKS or /VOLUME.NAME/APPLEWORKS, of a directory.

(3) The application name. This must be a full or partial pathname of the application program itself. The program must be of filetype S16, SYS or EXE. Thus

```
                    APLWORKS.SYSTEM
```

and

```
                    */APPLEWORKS/APLWORKS.SYSTEM
```

are valid examples, but for the former to be correct the prefix (item 2) must be such that adding this application name to it results in the correct FULL pathname. In the example, APLWORKS.SYSTEM is a valid application name if the prefix is */APPLEWORKS.

(4) The startup. Usually the startup is null. It is the only part of these four-part specs that can be left empty. It can be used to communicate various things to the application program. (However, the application must be listening - i.e., it must be something that the application was programmed to look for.) In GS/OS, the startup is usually the pathname (full or partial) of a file you want the program to open (load) automatically. It could also be some special instructions; some of the ProSel applications use it that way, and the program's documentation will tell you what the program looks for, if anything, in the startup.

If the application is BASIC.SYSTEM, then the startup is the BASIC program you want run by BASIC.SYSTEM. If you do not give a startup when you run BASIC.SYSTEM, then BASIC.SYSTEM looks for and, if found in the prefix #0 directory, runs the program called STARTUP. Thus, the startup, if any, in the ProSel specification is substituted for the default startup - the file called STARTUP.

Some other programs work similarly. PROSEL.SYSTEM, which is the boot program for ProSel-8, uses the startup in a similar manner, running the program given in the startup as the first thing it does. If a startup is not specified then it does not run anything but just goes to the ProSel-8 screen. In this case, in distinction to BASIC.SYSTEM, the startup must be a SYS file.

In the MANUAL mode of the editor, you specify these four things or just three if you don't want to give a startup. The prefix does not have to end with a / character.

ProSel uses some conventions in addition to the general pathname syntax discussed above. The ? character is used by ProSel to stand for the current volume name with a slash on both ends. Thus ? usually gives the same result as */ but not always, since the current volume name (which means the name of the volume from which the file PROSEL.SPECS was last loaded) does NOT have to be the boot volume. Use of this character provides transportability of application specifications as well as being a space saver. We recommend its use ALWAYS in any full pathname. REMEMBER that the ? includes the / on both ends, so that ?APPLEWORKS would be a correct prefix, but ?/APPLEWORKS would not since it expands to /VOLUME.NAME//APPLEWORKS and the two adjacent slashes are not legal.

Similarly, the character] anywhere in a pathname is expanded to BASIC.SYSTEM. Thus the two characters ?] will be interpreted by ProSel as /VOLUME.NAME/BASIC.SYSTEM.

The ? and] conventions are legal in both ProSel-16 and ProSel-8.

ProSel-16 also uses some other special notation in the prefix field as instructions to do certain actions before running the program that has been selected:

(1) If the character = alone follows the prefix, then the Apple-IIgs will be switched to slow speed before the program is run.

(2) If the character = follows the prefix, and itself is followed by one of the digits 0 to 7, then the following actions are taken before launching the application:

=0 purge memory.
=1 purge memory and force reload of ProSel on application exit.
=2 purge memory and do not return to ProSel-16.

```
     =3 purge memory, do not return, and force reload next time
        ProSel-16 is started.
     =4 set TransWarp-GS to slow speed (if card exists).
     =5,6,7 same as =4 together with =1,2,3 respectively.
        (The TWGS will be reset to fast speed on return to ProSel.)
```

NOTE: ProSel-8 uses some similar "=" notations that mean entirely different things. These things do not apply to an Apple IIgs and so there is no real conflict between the two.

ProSel-16 supports a convention that instructs it to go to the main menu and highlight a particular item there. To add such an item to the screen, you must give ? as the prefix, the command letter (eg, U for Utilities) for the "application" and have an empty startup. That is:

```
   Screen title:  Utilities-16  (or anything)
   Prefix:        ?
   Application:   U             (main menu command char.)
   Startup        <none>
```

Finally, both ProSel-16 and ProSel-8 support some conventions that instruct ProSel to read a directory and display its contents:

```
   Screen title:  Games directory (or anything)
   Prefix:        ?GAMES          (full pathname of directory)
   Application:   /               (special instruction)
   Startup        <none>
```

That example will read the directory given by the prefix and display all executable files (S16, SYS, and EXE) for selection.

Another example:

```
   Screen title:  BASIC programs  (or anything)
   Prefix:        ?BASIC          (full pathname of directory)
   Application:   ]               (Pathname to BASIC.SYSTEM)
   Startup        ]               (special instruction)
```

That example will read the ?BASIC directory and display all BAS and BIN type files for selection. Note that the Application path must be a valid path to BASIC.SYSTEM - it does not have to be just the] character as shown - BASIC.SYSTEM could be in a completely different directory and this would still work if the Application path is a FULL pathname leading to BASIC.SYSTEM. The startup here is the special instruction telling ProSel that this is a request to read a directory and not one to run a program.

After using the editor to edit the application list or adding to it, press ESC to get out of manual mode and another ESC to go from the editor main screen to the editor menu. There you will have several options. Ordinarily you want to press P to save the revised PROSEL.SPECS file. There are also options to load and save screens, which will be dealt with later.

Importing ProSel-8 application lists:

Many will want to transfer their present ProSel-8 application list to a ProSel-16 list, and a very easy procedure has been provided for doing so. Enter the ProSel-16 editor and press ESC to get to the editor's menu. Select "Load a screen", even though it is not a "screen" that you wish to load. When asked for the pathname to load, give the full pathname of the PROSEL file (eg, */PROSEL). Edit it if you want, and then when you want to save it as a ProSel-16 application list use the P command, NOT the S command, to save it. This will write over the current PROSEL.SPECS file thereby converting the ProSel-8 specs list to a ProSel-16 specs list. NOTE that the internal format of the ProSel-8 PROSEL file, and the ProSel-16 PROSEL.SPECS file is very different, so it will NOT work to simply copy the PROSEL file and rename it PROSEL.SPECS. The editor automatically adjusts the format of the file when it loads and saves.

Similarly, as will be remarked again when we talk about the "screens" feature, you can import a ProSel-8 "screen" by loading it into the editor, and you can resave it back either as a ProSel-16 "screen" (again a different format) with the S command, or as the main ProSel-16 application list with the P command. The editor does not support conversions the other direction.

Screen blank and/or slide show:

After a certain amount (user specified in the parms) of inactivity the screen will blank to black. It will come back at any keypress.

Alternatively, you can select a "slide show" to automatically appear after the given amount of inactivity. This is selected through the "modify parameters" function. The slide show expects to find its "slides" in the prefix #31 directory. This should also be set up in the parameters. The slides must be super-hires files (either unpacked of filetype $C1 {PIC} and auxtype 0 or packed of filetype $C0 {PNT} and auxtype 1, and of length at most $8000). [Also see the PackPic transient command and help file.]

At any time during a slide show, the ESC or CLEAR keys will abort the show and produce the normal screen blank mode, at which time another keypress will produce the ProSel-16 screen. The RETURN key during a slide show will produce the ProSel-16 screen directly. The space bar will freeze the current picture until another key press. Any other key will cut short that particular slide, and bring up the next one. When the sequence of slides is used up, the show cycles to the first slide again. In the parms, you can set the number of repetitions of the slide show done before passing to the screen blank, or you can specify that it is to go on forever, until an appropriate key press.

Help:

The help facility that is built in simply reads the prefix #7 directory showing all the TXT type files and allows you to look at any of them. It does not require a complete filename from you, but will match the characters you input against the same number of characters from the filenames and read the first file which matches to that point. The help files are standard text files so that you can edit them or write some of your own. A carriage return is expected at the end of each line, and you MUST take care that this condition is met. A linefeed (^J) character will cause the output to pause and the "Press a key for more, ESCAPE to quit" message to appear.

Show prefixes:

The Show prefixes selection from the main menu (or OA-P <RTN> from the screen) displays the values of the GS/OS prefixes, and the device names for the first 22 disk devices. It is advisable to get aquainted with the correspondence between device numbers and the disk drives associated to them. Show prefixes also indicates the characters that ProSel-16 puts on the screen to indicate the type of file listed by one of the "display directory" features (accessed either by a number key from the screen, or by one of the special specifications designed for this purpose). Finally, Show prefixes prints thermometers showing the relative usage on up to 6 disk devices other than 5.25 inch drives, and memory usage. The memory usage thermometer will probably have a blank area without the division marks, which indicates the "maximal memory block size".

Refresh screen:

The refresh screen item on the main menu simply rereads the PROSEL.SPECS file from the main directory of the disk corresponding to the current prefix #0. This can be used to get rid of a "screen" that has been loaded over the main screen. Its primary purpose is to switch to a PROSEL.SPECS screen on another disk, and you can do that by first accessing the disk with a number key on the main keyboard and then giving the refresh command OA-R <RTN>. ProSel-16 will continue to use that screen until changed by another refresh command to a different disk, or until it cannot find a PROSEL.SPECS file on that drive. In the latter case it will poll all your disk drives until it locates one such file. If none can be found (or if the file is somehow damaged) then an error message will be given, and you will have the option of going to shell command mode. (One of the primary purposes of shell command mode is for just this back door. It allows you to get out of bad situations gracefully.)

Shutdown:

There are two items on the main menu called Cold shutdown and Warm shutdown. These shut down the GS/OS operating system as recommended by Apple as preparation for turning off the computer. They should also be used, if possible, if you need to reboot for some reason. The difference between Cold and Warm is that the Warm shutdown will not resize or clear /RAM5, while Cold shutdown is the same as a power down reboot - it will clear /RAM5 and it will pick up any change in the size of /RAM5 that you have made in the control panel.

Exchange screens:

This option in the main menu is equivalent to the PROSEL.CYCLER program in ProSel-8. Selecting this item will bring up a dialog with choices for changing the present application list to another one that you have previously made. These application lists are versions of the PROSEL.SPECS file and must be in the main directory with that file, and must be named PROSEL.SPECS.2, PROSEL.SPECS.3, etc. To make such a file, copy PROSEL.SPECS, using the Utilities, to the main directory (where it is now) and when given the option of renaming its image, select one of the names PROSEL.SPECS.2, etc. Then to edit one of these, select the Exchange screens function, and select one of those new lists. What happens is that the PROSEL.SPECS file will be renamed PROSEL.SPECS.1 (or whatever is the first "missing" file) and the selected copy will become the new PROSEL.SPECS file. Thus, when you go into the editor, that NEW file is what you will be editing.

By means of a special type of application specification, you can set up a screen item that goes directly to one of these extra

ProSel main screens, without going through the Exchange screens item on the main menu. This specification will cause a change to the extra main screen number 3:

```
   Screen title:  Word Processors (or whatever)
   Prefix:        ?               (does not matter)
   Application:   X               (meaning the Exchange item)
   Startup:       3               (= the desired main screen)
```

Modify parameters:

There are a number of user modifiable parameters in ProSel-16, including printer parameters for various parts of the program, names for the "user" filetypes, the correct pathname for the BASIC.SYSTEM file that is needed by the File Finder in order to launch BASIC programs from there, selectable pathnames for the prefixes 0 through 31, etc. Selecting the Modify parameters function from the main menu will take you through all these choices, presenting the present values for approval. You can press return to accept the shown value, or you can type in the value you want to change it to. In many of the changes, you do NOT have to press return after entering a change. At any time, you can press ESC to move immediately to the end of the routine. At the end of the routine, you are given the choice of saving the changes (these go into the file PROSEL.PARMS which is in the prefix #1 directory - usually */SYSTEM), or just using them on a temporary basis, or of cancelling all the changes you have made.

Boot programs:

One of the user-settable parameters in Modify parameters is the option of specifying the name of a program to launch automatically upon boot, bypassing the ProSel-16 screen. In fact, Modify parameters also allows you to set its default prefix (prefix #0) and its startup (such as a file for it to open).

If this boot program is very large then there is a chance that it will force ProSel-16 to reload when it quits. This could easily lead into a cycle where the program gets run again automatically after it quits back to ProSel-16, which, having been purged, has forgotten that it ran this program just previously. To avoid this problem, ProSel-16 keeps track of the day the boot program was last launched in this manner and will not honor a relaunch on the same day IN THIS MANNER. (Direct selection of the program from the screen is not affected by this.) [One of the user modifiable parameters can defeat this.]

In addition, there is a provision for a keyboard action to force ProSel-16, on boot, to ignore the boot program specification: just hold down the CONTROL key when booting (it has to be down when ProSel-16 is booting - say from about three quarters of the way along a boot until the screen clears). There is also a key that will FORCE recognition of the boot program (overriding the day check): just hold the SHIFT key down in the same manner.

NOTE: The CONTROL and SHIFT keys have a similar function when running a program normally from the screen: CONTROL means to ignore the startup, SHIFT means make a certain special startup.

The File Finder:

When you select this option from the main menu, you will be asked for a device number to search. The boot volume is always device number 1, and the others can be found by using the Show prefixes option at the main menu. After you give the device number, you will be asked for a file name (or a string - see below). The find file routine will then search through the entire volume for this file. When found, the name of the directory containing the file will be shown and the catalog data concerning the file will be displayed. At this point you will be asked to press a key. The RETURN key will cause the search to continue for another file of the same name in another directory. The ESC key aborts the routine. The "D" key produces a file dump in hex and ascii. You may select printer output of this. The "T" key "types" the file. In this case control characters other than carriage return will be shown in inverse, or ignored if output is to a printer. If the file is a BAS type file then you can also use the "L" key to produce a formatted program listing in which all statements are shown on separate lines and loops and conditionals are indented.

When using any of these modes you can make the listing pause or step by pressing the space bar or you can abort the listing with the ESC key.

If the file is an S16, SYS or EXE file then you also have the option of running the program with the "R" key.

You can use ? anywhere in a file name as a wild card for any single character, and you can use = as a wild card for any string of characters. More than one ? or = can be used.

If the file is not found or if a disk read error (or other problem) occurs then you will be told of this and asked to press a key, after which you will return to the main ProSel-16 screen.

You can also use this to run a BASIC program. For this to work you must have specified, in Modify parameters, the location of the BASIC.SYSTEM file. When a BASIC program is run this way, the prefix will be set to the directory containing the program. This directory does not have to contain BASIC.SYSTEM.

The file finder can also find files containing a specified text string. To do this you simply type the string at the "filename or string" prompt, but precede it with a quotation mark (single or double).

The program will then ask for a directory to search. If you just press RETURN at this point, the entire disk will be searched. If you specify a directory then only that directory, and its subdirectories, will be searched.

You can then specify a filetype to search. For example, if you know that what you are looking for is in a file with filetype TXT then typing TXT at this point will limit the search to those files, substantially speeding up the search. If you want to search all files then just press RETURN at this prompt.

Then you are asked for a minimum revision date of the files to be searched. If you know that the item you want is in a file that was revised since a certain date, then giving that date will speed up the search. If you just press RETURN then dates will be ignored.

Finally, File finder asks if it should display the filenames of the files it is currently searching. Answering N will slightly speed up the search if you are doing a search through an entire hard disk, but Y makes it more interesting to watch.

When the string is found, the program will display the catalog information about the file, and will display the portion of the file surrounding the found text, and highlight the found string. The string will be around the middle of the screen, unless it is very close to the beginning of the file, in which case it will be closer to the top of the screen.

You will be told the "byte offset" of the found string in the file, and asked if you want to continue the search or not, and, if so, whether you want to skip to the next file, or continue with the present one. The byte offset can be used with the Type function to quickly go to the location of the found string, if you do a subsequent file search for the found file.

The string search is not case or "high bit" sensitive.

Information desk:

This module will print the entire tree structure of a selected volume or directory. The output can be sent to the 80 column screen, to a printer or to a disk file.

There are five main parts to Info Desk that give you different types of documentation of the files on your disk:

1. Catalog.

   This mode prints a tree structured catalog of the entire volume. Subdirectory contents are indented two spaces. All the usual catalog information is printed except the time of day and the access (locked) status. Access status is printed if the line length is set to 90 or more. The time of day in the dates is printed if the line length is set to 102 or more.

2. Block usage by files.

   This mode tells you just what blocks on the disk are used by each file. The printout gives the number of blocks in each file (as in mode 1). Then comes the Index field. This contains the number of the index block. This field is empty for directory files and for seedling files (which have no index block).

   Next come the actual data blocks. If two numbers in this list are separated by periods then they represent a range of blocks all belonging to that file.

   In case of a tree file (length > $20000 = 131072) the first entry in the Index column is the master index block indicated by " <== (Master index block)" next to the block number. After this, on subsequent lines, are the index blocks pointed to by the master index block, and their associated data blocks.

   At the end there is a "fragmentation" count given. This is a count of blocks that are not in line with the rest of the file and gives a rough indication of when an optimization might be called for.

3. File usage by blocks.

   This mode is the "reverse" of mode 2. It shows which files belong to the blocks on the disk. Most of the disk operations are done prior to any printing, so be patient. The printout consists of ranges of blocks in numerical order followed by the file name (the full pathname less the volume name) of the file which uses these blocks.

4. Bit map.

   This mode prints the volume bit map. Used blocks are shown with "x" and free ones with ".", but these are user definable by the Modify parameters option.

5. Tree structure.

   This mode prints a tree diagram of the volume.

There are also sixth and seventh options which are the same as options 2 and 4 except that they have a super-hires graphic display. They are available only if you choose output to the screen (slot 3). The pixel size depends on the volume size being mapped. A volume of 16MB or over will have the smallest pixel size. 3.5 inch disks will have a large pixel size resembling low resolution graphices. The TAB key will toggle between the graphics display and the text display. Since there are only 64000 pixels in super hires compared to a possible volume size of 65535, the last 1535 blocks will be ignored in the display whether they are used or not. The graphic mode #6 can be used to observe the level of fragmentation of a disk by how much jumping around is done by the plotting. Note, however, that the optimizer puts all directory blocks at the beginning so that there will be normal jumping between the current location and the first couple of lines of the display. Also, since tree files go at the end, some display of large numbers of blocks near the end of the display will be normal for optimized volumes. Directory blocks are plotted in white, and other blocks in 14 other colors changing with each new file. The area of white at the bottom of the screen represents block numbers beyond the volume size and thus makes possible a visual representation of the free area on the disk.

At any time during printout the space bar will stop, then step, the process, and ESC will abort it. When printing to the screen is finished, the program waits for a keypress before clearing the screen and asking if you want to do another.

If you choose the option to output the data to a disk file the file will be placed in the prefix #5 directory. You can use Modify parameters to set this prefix to anything you want, and you can change it while in ProSel by using the shell prefix command. The filename used when the Catalog option (#1) is in effect is "CAT." followed by the volume name. (If this is longer than 15 characters then it is truncated.) If this file already exists it will be overwritten unless it is locked. If it is locked the file finder will abort. Similarly, for the options 2 to 5 the file name is "LOC.", "BLK.", "BIT.", or "TRE." followed by the volume name.

Backup/Restore:

This is a fast full volume or incremental backup facility. You can backup the files on any volume or those in a subdirectory, and can choose to backup all the files, only those changed since the last backup (determined by "backup bits") or those whose modification date is later than some specified date. It is "file oriented" so that the volume size can be changed between backup and restore. At this time, 3.5 inch disks are the only accepted backup media.

In order to maximize the backup speed, the program requires enough free space to hold the contents of a 3.5 inch disk. There is an option in the parameters that will cause it to use only 400K of memory and partition the backup disks into two parts. People with only a 1MB memory card should use that option. If this memory is not available, you will be told at the start. The backup will operate at about 2/3 MB per minute, but this depends on the speed of the media used. Restoring is slower, about .4 MB per minute. Choosing the verify option, will almost double the backup time.

There are some files that the program will not back up. These are read-disabled files, bad block files, "pascal area" files and files that are very sparse. The reason for the latter is that the program cannot efficiently ignore the zero blocks in a sparse file and such files may span many backup disks. These files should be copied individually to another disk for backup. The skipped files are displayed before the reading of files for each backup disk. If backup bits are cleared then these files will again be displayed at the end of the backup process. Modify parameters lets you set the "sparseness ratio" above which sparse files will be ignored. A value of 2 or 3 or 4 means that files with over 1/2 or 2/3 or 3/4 empty blocks will not be backed up and similarly with other values. We suggest a value of 4 or 0. Setting this parameter to 0 defeats this feature altogether.

To use Backup/Restore with two alternating backup disk drives you must first use Modify parameters to set one or two 3.5 inch disk devices to use for backup purposes. If you only want to use a single drive for backup, make both of these seletions the same device. Since there is considerable time between ejection of the disk and its next usage, single drive backup is almost as convenient as is use of two drives. If you do not set the backup devices in the parms, the first Apple 3.5 drive (NOT Unidisk 3.5) will automatically be searched for and used.

One should make a full volume backup every week or two. You can use this facility for that or the ProSel-8 backup. If you use the latter, you should choose the "clear backup bits" option. The ProSel-16 backup always clears backup bits of the files it backs up, except for the "compound" backup bits mode. This is done only at the end of a successful backup.

The "compound" mode is identical to the "backup bits" mode except for the clearing of backup bits. If you use this mode then every backup since the last full backup will backup all changed files since that last full backup. This allows you to use only two sets of backup disks - the full backup set and the incremental set. You must not mix the three incremental modes - choose one to use and stick to it.

Using separate backup disk sets for each day, an incremental backup by "backup bits" is recommended at the end of every day. If a restore should be necessary, you can go back to any one of these sets of backup disks (including the full backup) and do a restore, followed by restores of all subsequent incremental backups in the same sequence in which the backups were made.

Backup disks are named BU.01, BU.02, etc. Best results will be had from media formatted with 2:1 interleave.

You do not have to reformat or erase the backed up volume before doing a restore if the volume has no present problems. However, if there is some problem on the disk (damaged files or directories), then you should erase the disk before a restore. The restore function, when done on all files of a volume directory will ask you if you wish to erase. If you have a bad block file, then you should not use the erase option. Instead erase the disk by another means and use Volume repair to rewrite the bad block file. When you do this, you should first boot the ProSel disk from a 3.5 inch drive.

> NOTE: Erasing the disk will also remove the protection of the DOS.MASTER partition if you have one. It will not harm the data in the partition, but that area will be open to subsequent overwrite. To reprotect it you can use Zap to zero out the bit map (starting at block 6 and using one block for every two megabytes), and then use Volume repair in F mode to free up all the blocks not in the partition. This requires the DOS.3.3 file to be on the disk (see the doc for the Volume repair module).

When you enter the Restore routine, you will be informed of the directory that was backed up onto the present media and of the mode of backup (all files, by backup bits, or by modification dates).

Some read/write errors are not recoverable and will force an abort of the program. Most, however, will allow a retry or a "continue".

In the latter case the problem file is just ignored.

If the program wants a backup disk and there is no disk in the backup device, you will be prompted to insert the proper disk. If there is a disk but it is not recognized as a ProDOS disk then you will be prompted about formatting it. If there is a disk in the backup device but having the wrong name, you will be asked if it can be "destroyed". If you answer N, you will be prompted for the correct disk. If you answer Y then the disk will be written to and will inherit the proper backup disk name. You can just replace the disk with the correct one and answer Y to this.

The "selective recover" allows recovery of individual files from a single backup disk. It will ask for one of the disks, will read it, and display a list of backed up files on it. Use the arrow keys to scroll through the list, the space bar to mark files for recovery, and press RETURN to start recovery of the marked files. Press ESC to abort. If the last file on the disk is marked then the program will also read the next backup disk in case the file is continued there. While selecting, the ^A key toggles all marks.

> NOTE: The BACKUP and RESTORE programs in the UTIL directory are ProSel-8 programs. The incremental backup/restore discussed here is accessed through the main menu (ESC from the screen) or directly from the screen as shown below.

Backup/Restore can be automated by putting the desired key sequence in the startup of a specification for backup. The first character, selecting Backup or Restore can be followed by a semicolon to separate it from the following directory name. Otherwise a semicolon will be interpreted as a RETURN. Similarly, a "<" character in the input of the directory name will move the cursor to the start. For example, you can set up this specification:

```
Screen title:   Backup, incremental
Prefix:         ?
Application:    B
Startup:        B;HARD1;C
```

and selection of that from the screen will go into the backup routine, provide the directory name HARD1, select the compound backup bits mode and begin the backup.

Similarly, the following specification will cause a full backup of the /HARD1 volume:

```
Screen title:   Backup, full
Prefix:         ?
Application:    B
Startup:        B;HARD1;A
```

A disk access error during backup or restore will turn off the automatic mode.

The "clear backup bits" option is intended for use by people using another backup method which does not clear backup bits (such as SiderFile) for full backups and the present system for incremental backups. In this case you should use the clear backup bits option immediately after doing the full backup. This is automatic when using the present system for doing the full backup.

The "directory" option lets you see what files are on a particular backup disk. (Selective recover does the same but not as fast.) Pressing RETURN at the end causes a return to the start of the backup program instead of exiting. ESCAPE exits. Pressing "*" when the directory is on the screen will cause it to be sent to the printer.

The "update" option restores only those files on the backup disk that either do not exist on the destination volume or have a later modification date than on the destination volume. This facility is intended for using backup disks to transfer files from one hard disk to another one. It could also be used to recover deleted files from a set of backup disks made before the deletions.

The "script" option lets you backup via a list of files and directories. To access this option, just give the name of the script file when asked for the name of the directory to be backed up. The prefix upon entering the backup module will serve as the prefix throughout the backup. The script file must be a standard text file. (It can be written, for example, by the RECORD function in the shell, or by any text editor that can write standard text files). It must contain the names of the files to be backed up. Each name must be a PARTIAL pathname (that will be added to the prefix) and must be terminated by a carriage return. The script can also contain directory names, provided they are also partial pathnames, in which case ALL files in the given directories and their subdirectories will be backed up. You can use any of the backup options of all files, files with backup bits set (compound or not) or files revised after a given date.

Since the prefix on entry to the scripted option is crucial, I suggest that you do not use this option via the keyboard, but rather from a screen specification such as:

```
                Screen title:   Backup by script
                Prefix:         /HARD1/MYDIR
                Application:    B
                Startup:        B;HARD1/PROSEL.16/BACKUP.SCRIPT;A
```

The script file BACKUP.SCRIPT could contain, for example, the lines:

```
                WP.FILES
                DB.FILES
                DATA.FILE1
                DATA.FILE2
```

Then selection of this item will automatically backup all files in the directories /HARD1/MYDIR/WP.FILES and /HARD1/MYDIR/DB.FILES and also the single files DATA.FILE1 and DATA.FILE2. If the specification Startup had ended with C instead of A then only the files, among those, that have changed since the last full backup will be backed up.

Note that the script file name given in the Startup is a full pathname but without the "/" at the start. This is necessary because the directory name prompt accepts only full pathnames but supplies the first "/" automatically.

There is an option (in Modify parms) to use error correction on the backup disks. This will correctly restore from backup disks that have a few unreadable or corrupted blocks. (If the error correction is unsuccessful, an error $27 results.) The error correction has a penalty of using about 20% more backup disks, and the corresponding time necessary to write to those disks. I think the added reliability is worth it, but some may prize speed more and be willing to live a little more dangerously.

Success of the error correction depends, of course, on the number of bad blocks on the backup disk. If the bad blocks are randomly distributed then the chances of a successful recovery are given by the following table:

Number of bad blocks	Chance of successful error correction
3	99%
5	97%
10	87%
15	72%
20	55%
30	25%
40	8%

50	2%

These statistics were theoretically derived. Actual tests indicate somewhat better chances, probably due to the fact that non-randomness would tend to improve the chances. For example, several bad tracks in a row would always be recoverable unless one of them is the first track on the disk. (The first three blocks contain vital information and must be readable for error correction to succeed.) Similarly, you could touch the disk surface with a strong quarter inch bar magnet and have a 100% chance of recovery, unless you happened to hit one of the first three blocks. In the latter case, if you get the message "Insert BU.xx..." and the correct disk is in the drive, press the option key to force accepting of the disk (only in error correct mode).

There are actually two varieties of error correction in the backup program, one for dealing with bad blocks and another for dealing with corrupted data. Thus it is possible to see the message "Backup disk bad - attempting fix..." twice in succession. If you are desperate and willing to live with corrupted data, you can defeat the second of these integrity tests by putting the byte $FF in the first byte of block 0 of the backup disk. The first test can be defeated only by copying the good blocks to a totally readable disk.

Through Modify parms, you may also elect to skip certain filetypes in backing up. You might choose, for example, to skip the system files and perhaps application files (SYS and S16 types) since those would be easily replaced. This would speed up the backup process and require fewer disks, but it would also make a full restoration of the disk more difficult. You can also elect to skip "hidden" files, a feature that lets you skip individual files in the backup.

Zap (Block Warden):

The Zap module is a block editing facility. When "Zap" is selected from the menu, block 2 of the boot disk will be read. The module is entered in "R/W mode". Only one memory page of a block is shown at any time and you must use the "\" key to flip between the two pages of a block in both R/W mode and EDIT mode. (In EDIT mode however, moving the cursor will automatically adjust the page displayed.)

R/W MODE

In this mode of Block Warden the arrow keys read the next or previous block. The down arrow and up arrow read then next or previous PAGES.

Pressing the right or left arrow in R/W mode while holding down the Option key will act as if the arrow is pressed 10 times. The Apple key will do the same but 100 times. This is active whether in follow mode or not. Note that if following a sparse file, the next block read may be beyond 10 or 100 blocks farther in the file since the arrows skip over missing blocks in sparse files. The up/down arrows are not affected.

The Q key will ask if you want to quit.

The R command allows you to specify the next block to be read. Block input is in hex unless you backspace over the $ sign, at which point you can enter a decimal number.

The W command will allow changing the block. (Just press RETURN for no change.) Then it will ask if you really want to write block as an extra safety to prevent accidents. Any write command will force a warm shutdown when you quit from Block Warden since otherwise GS/OS may get confused and create problems on the disk.

The C command allows you to change the disk device being accessed. Note that this command will not read from the disk, so that you can transfer a block from one disk to another by use of it. Subsequent reads, however, will come from the new disk.

The P command allows you to specify the prefix. This is mostly for use with the F and I commands.

The F command asks for a pathname (full or partial) to be followed. After issuing this command, the file name being followed will show at the top of the screen and the arrow keys will move between blocks OF THE FILE. Hitting the ESCAPE key (and some other commands) will cancel the follow mode. This facility supports all file types including directory files and sparse files. It reads only the data blocks of a file, so it cannot be used to look at the file's index blocks. While following a file you may go to edit mode, edit, return to R/W mode, write a block, and continue to follow using the arrows.

The I command asks for a file name and then shows the file parameters which appear in the directory entry of the file in a form that is easy to read. The bytes on the left give the offset location of the data in the block buffer. When you press a key the program reverts to the block in effect when the command was issued. If the file is a subdirectory, however, the header of the subdirectory is shown after the first key press. You must note the block number containing this information if you are going to want to use the editor to change any of this information.

The L command disassembles the current buffer contents. [In follow mode with a TXT file it lists in ascii instead. This can be forced in follow mode for any file type with the " command.] The ascii equivalents of bytes appear after the disassembly line. A total of 40 lines are listed on each page. You are asked for a starting byte (0-1FF) to start the disassembly on. If you just press RTN this defaults to zero. During disassembly the right or up arrow or RTN produces the next page of the disassembly; another key cancels this mode. If you are following a file then the right or up arrows pass to the next block of the file when the buffer listing is done and that block is listed. While in the List mode you can dump the screen to a printer by typing Apple-D. This assumes the printer is on and in slot 1. (If a previous Dump command was issued and you selected a different slot for the dump then that slot will be used instead.) The disassembly supports the whole 65816 opcode set. The "M and X flags" attempt to follow the program logic, as in the Merlin-16+ assembler. At the start of each page these flags can be reset by using the RTN key instead of the arrow keys and pressing, at the same time, the Apple key to set M=0 and/or the Option key to set X=0.

The D command dumps the buffer contents to a printer which is assumed to be in the slot of the number key you press when you are prompted to turn the printer. You can abort the command by pressing the ESCAPE key.

Finally, the ^ command sends a list to the printer of all possible "index blocks" on the disk (starting with block 7). This is intended as an aid in a last ditch attempt to repair a blown directory. It would be better if you have an up to date list made by INFO.DESK. Also note that if block 2 is bad (so that the volume name is shown as "?") then this command will not work - it will print ALL the blocks on the disk. Thus you must repair block two first to the extent that it has a valid volume name and the correct number of blocks (bytes $29,2A of block 2). The list printing can be aborted by pressing the ESC key. The list will include some blocks that are not really index blocks, mostly partially full blocks at the ends of files.

EDIT MODE

In edit mode the arrow keys move the cursor (shown in inverse). Any key other than a control character will be regarded as a change to the editing buffer. (In hex mode it is disregarded if not a valid hex digit.)

The ESC key returns to R/W mode. The ^X key cancels any changes you may have made on the current block. (This is done be rereading the block.) The TAB key toggles between hex and ascii editing modes. You can tell what the current mode is by seeing where the cursor is placed. In ascii editing mode, the high bit of a typed character will be off unless you press the Apple key at the same time, in which case the high bit will be on. Control characters can be inserted into the buffer only in hex editing mode.

The ^F key is a lead-in to character find mode. If the cursor is in the hex portion of the screen then the cursor will disappear and you are expected to type two hex digits. (An invalid digit causes this mode to be canceled.) The resulting byte then becomes the "find character" and the next occurrence of it in the buffer will be found and the cursor moved there. (If none exists the cursor returns to the first byte and the "Find mode" message is erased.) After this first find (that is, when "Find mode" is on the screen) subsequent ^F commands find further instances of the find character in the buffer. This mode can be cancelled with the RTN key.
If the ^F key is issued when the cursor is in the ascii portion of the display then an ascii find character will be gathered. (High bit of the find character will be off unless the Apple key is pressed when the character is typed.)

The ^S key selects global search mode. When used the first time this brings up a request to input a search string. If the string you give begins with "$" then it will be taken as a hex string for the search. Otherwise the string is taken as an ascii string.
If FOLLOW mode is in effect then only the followed file will be searched for the string, otherwise the entire disk, from the present position, will be searched. The ascii search is not sensitive to the high bit of characters, nor is it case sensitive. (Thus, for example, a search string of Junk will find both "JUNK" and "junk".) You can cancel a search in progress with the ESC key while the disk is reading, or during input of the search string. Strings that cross block boundaries will be found by this facility. After the first instance of the search string is found, others will be searched for upon pressing the ^S key in edit mode. Any block read with the R/W mode Read command will cancel the search mode. You can also cancel it by pressing RTN in edit mode. The search mode with the same string can be reinstated by typing another ^S command in edit mode and just pressing RTN when the default string is shown.

If the Volume name at the top of the screen, is shown to be "/?"

this means that the program encountered an error in trying to determine the name. This almost certainly means that there is something wrong with block 2 (the first block shown). It could also mean, however, that the disk is not a ProDOS disk.

CHANGING STARTUP NAMES

BLOCK WARDEN has a built in feature to simplify entering STARTUP names in SYS files. To change the default STARTUP file name in a SYS file, follow these steps:

1. In R/W mode, press F to follow a file, and specify the appropriate file name. (You may want to use the P command first to set the prefix.)

2. When the first block of the file is read, use the E command to go to edit mode.

3. Use the right arrow key six times to move the cursor to the startup position. At this point a box will be displayed asking for the startup name. (If not, then either the file is not of SYS type, you are not in follow mode, or the file does not support startups.)

4. Type the startup name ending with RETURN. When done you will see the name you entered has been placed in the edit buffer with the appropriate length byte supplied automatically.

5. Press ESC to go to R/W mode, then the W write command to write the buffer to disk, etc.

SUMMARY of BLOCK WARDEN commands:

R/W mode commands:

Q.......	quit to the ProSel-16 screen
E.......	go to edit mode
Arrows..	read next or previous block or page
R.......	read block (input)
W.......	write block (input)
I.......	get file info (input)
",L.....	list/disassemble buffer contents (input) (continue list with right/up arrow or RTN) (RTN with Apple keys operate MX flags)
P.......	set prefix (input)
F.......	set "follow" file name (input)
C.......	change device (input device #)

```
             D....... dump block contents to printer
             ^....... send index block list to printer
```

Edit mode commands:

```
             ESC..... return to R/W mode
             Arrows.. move cursor
             ^X...... cancel changes to buffer
             ^F...... find chr next typed in current buffer
                      (or continue find)
             ^S...... global disk or file search for string
                      (or continue search)
             RTN..... turn off find and search modes
             (Non-control keys are taken as buffer edits.)
             (Open Apple key after ^F in ascii mode or during buffer
             edits sets the high bit of the character typed.)
```

Optimizer:

CAUTION: This utility is extremely dangerous. If it misfires in any way or if you have a glitch in your system such as a slight ram problem or if you have a power outage while it is operating it can destroy the entire volume it is working on. You absolutely MUST have a full backup before using it. Destruction may also occur if the disk contains bad blocks or damaged data in a directory. (Bad block files are respected by the optimizer, however, and will not cause difficulties.)

I TAKE NO RESPONSIBILITY OF ANY KIND CONCERNING THE PROPER WORKING OF THE OPTIMIZER. YOU ARE ON YOUR OWN.

---> Make a BACKUP first !!! <---

The volume optimizer will put all directories at the beginning of the volume, and all files will have contiguous data blocks. This makes for much more efficient file access.

The optimizer will ask for a disk device number. It will then read and show the volume name in that device and ask if it is all right to continue.

The optimizer will take quite some time to operate. If you absolutely must abort the program, you can use the escape key. The program will take a few moments to react to this key because it will only quit at a time when the disk is "clean", and must also do some last moment writing to disk at this point. If you later want to continue, the program will quickly arrive at the spot where it stopped, for the simple reason that it has little to do before that spot. DO NOT interrupt the program with RESET.

The optimizer makes 4 "passes" numbered 0 to 3. These are passes through the directory, not the disk. The brief pass number 0 just reads data from the disk, and an error at this phase will not affect disk contents. Pass number 1 relocates the directory and is not very long. However, the program logic in this section is extremely complicated and any obscure bugs are likely to show up at this time. [After all, directories have not only forwards and backwards pointers, but also upwards and downwards ones. Moving one directory block takes a lot of housekeeping.] From this point, the screen will show the current block being processed. Pass number 2 relocates all files except for tree files. This pass takes the great majority of the time used by the program. Finally, pass number 3 relocates tree files, if any. All tree files will be relocated following all standard files. (This refers to the files themselves, not to their directory entries, which will not be changed.)

Although I have gone to great lengths to make the program as fast as I can, it will still take a long time to execute on any sizable hard disk. About 2 to 5 minutes per megabyte is typical for the first run. Later runs may take much less, depending on what areas of the disk have been changed and how extensive are the changes. It has been known to run in under a half minute per megabyte in some cases. In any case it is much faster than Beach Comber under ProDOS 8.

The optimizer goes through directories and files in the same order as the INFO.DESK program. Thus, to speed later uses of the optimizer, you can put directories containing files that can be expected to change, late in this order. (I.e., late in the main directory.) Put stable directories early. Of course, this may not be practical for all people and it may not speed things up all that much anyway.

If the optimizer encounters a problem it will shut down and display an error message. Some error messages like Block Read error are self-explanatory. Others like Block Reference error are rather technical and just generally indicate some defect in the volume (or may be caused by a hardware problem). One error that seems to occur more than others is a "Block in use" error. This error occurs only on the initial pass before any writing is done to the disk. It indicates that there are two files, or a file and a directory or bit map block, that contain the same block. Thus one or both of the files has been corrupted. This is a serious error and this condition should be dealt with as soon as possible and as carefully as possible. Do not write to the disk until it is cured. Volume repair should report the same error as "Block in use". That will indicate one of the two bad files, and Info.Desk can be used to determine the other file. Such files MUST NOT BE deleted by ordinary means. Use the "problem file deletion" method described at the end of the documentation for Utilities.

> NOTE: There is a very serious problem in GS/OS that can utterly destroy a volume after an optimization. To avoid this problem, a shutdown is forced after an optimization.

Volume repair:

This is a directory repair utility corresponding to Mr.Fixit in ProSel-8. There are four modes: a Test mode in which nothing is altered on the disk, a Fix mode which attempts to correct defects found in the directory structure, etc, a Main directory mode which attempts to reconstruct the main directory and a Bad blocks mode which scans for bad blocks and, if desired, places them in a bad block file. The Fix and Main directory modes are dangerous and can change a bad situation to a worse one, so never use them without first using the test mode and never use them without an adequate backup that you can fall back on.

The program tests and, if fix mode is active, fixes the following defects on any ProDOS volume:

1. Header pointers of all active files.
2. Parent pointers and parent entry numbers of all subdirectories.
3. Backwards directory links.
4. Used blocks marked free in the bit map.
5. Illegal characters in file names.
6. Entry length (the program assumes this should be $27).
7. Number of entries per block (assumes this should be 13).
8. File count in each directory.
9. Incomplete deletes (deleted files with non-zero "name length").
10. Incorrect directory storage types.
11. File and directory block counts.
12. File and directory dates and times.

In file names (point 5) lower case characters are converted to upper case, high bits are stripped, and other illegal characters are replaced by periods. Illegal dates and times are zeroed. [Note that even though GS/OS now supports lower case in displayed file names, the names in the directory are still all upper case.]

The following items are checked and reported, but not acted upon:

13. Blocks used by two or more files ("block in use" error).

14. Block number out of range (past volume size).
15. Unknown storage types.

Some errors result in files or directories being skipped over. You will be told if this occurs. It happens because of information damaged in such a way that the situation cannot be handled, or the damage is such that the supposition is that the remaining data is invalid.

Some block read/write errors cannot be handled and will result in termination of the program.

Sometimes in the error printout you may see a directory name printed twice. This is not a bug; it indicates that the correction refers to the directory "header" rather than the "parent block".

The utility will also look for blocks that are marked used on the volume bitmap but are not used by any file. You will be given the option, in Fix mode, of releasing these blocks. Sometimes areas of a volume are marked off without belonging to any file and in this case you should not ask that these blocks be freed. Examples are the UNODOS area on a Unidisk or a Pascal area. As a general rule, if the number of blocks indicated as being marked, but unused, is large, then you should assume that they are marked for a reason and should not free them. (UNODOS, for example, marks exactly 400 blocks.)

The most important block on a ProDOS volume is block 2. Since it is accessed much more often than any other block, it is also the most likely to be damaged. If it is, you will probably see a message that block 2 is too damaged for the program to function. There is a special provision for attempting a repair of the main directory, and it is accessed automatically when you request Fix mode and block 2 has extensive damage. You can also force this mode by selecting M at the Test/Fix prompt. You will be asked if the program should assume that the "bit map" is valid (default = Yes). Ordinarily you should select Y unless there is some reason to believe the bit map is damaged. (For example, if you are trying to resurrect the subdirectories after a disk has been "erased" by Utilities then the bit map will NOT be valid and you must select N at this prompt.)

When this mode has been selected by M or automatically because block 2 has been determined to be substantially damaged, then the program will tell you that this attempt is being made, and will give a few particulars along the way. Although the routine can resurrect most subdirectory pointers, it cannot do anything for standard (non-subdirectory) files in the main directory. While this routine is operating, every block on the volume will be read. This may take some time, so be patient. When it is finished, you will get the "Another?" message. Note that this routine does not do the other repair jobs mentioned above. You can run through the disk test again to check if those things are all right. You should realize, however, that this main directory repair routine expects most of the rest of the volume to be reasonably valid and normal. (For example, it assumes that the "bit map" starts on block 6 if the main directory header has been damaged. This is true for almost all disks, but not for some RAM volumes; this routine should not be attempted on a RAM volume or any other volume you suspect is organized in an unusual way.

If you select the Bad block mode then you will be asked if you want only to test or to fix. If you select test then the volume will simply be scanned for bad blocks and the results reported. If you select fix then there will be an attempt to place the bad blocks found in a bad blocks file. This file will be created in the main volume directory (so be sure there is room for it). If a bad block is a data block and not a directory or index block then several attempts will be made to read it and relocate it. If it cannot be read then a fake block is substituted and marked with the message "DAMAGED BLOCK".

Volume repair can be used to resurrect the subdirectories in the main directory and all other files that are not in the main directory. (It is not possible to retrieve the vital pointers for other files in the main directory, although those files may still be intact on the disk somewhere.) This can be used on a disk that has been inadvertently erased or soft formatted. (I.e., as long as the disk was not physically formatted). To do this, first use the option M to repair the main directory. Then use the T mode to check the types of remaining errors. (There should be a LOT of block free errors, and a file count error.) Then use the F mode to fix the remaining errors if there are no fatal errors. You must remember that this program makes decisions that sometimes may be inappropriate and may leave some undetected problems.

The volume repair can be automated by putting the desired key sequence after a semicolon in the startup position of a ProSel-16 specification for it, as in:

```
Screen title:   Volume repair
Prefix:         ?
Application:    V
Startup:        ;1F3
```

This example will select disk device #1, Fix mode, and screen output.

If you use my DOS.MASTER program, also put the pathname to the DOS.3.3 file in the startup (before the semicolon if you use one).

> NOTE: A Shutdown may be forced after repair of the boot volume (only if some actual writing was done), because GS/OS will get confused and destroy the volume if this is not done.

The "statistics" mode of Volume repair will analyze a hard disk (or any disk) for speed of access in both a "linear mode", reading successive blocks, and a "non-linear mode", reading randomly ordered blocks. The average time to access and read a block is reported for both of these. This mode is "test only" and never writes to disk. The two tests begin as soon as the S is pressed selecting that mode.

Utilities:

The utilities section is a filing and directory manipulation utility. Its main function is to do batch copying, locking, unlocking, and deleting of files from specified directories. Most of the routines are limited to directories containing at most 767 entries, but more than 110 will cause truncation of file names on the display.

Throughout, pathnames must be given in their FULL form. The TAB key moves the cursor past the next "/", or to the end of the name. Apple-TAB moves the cursor back to the preceding "/". In some routines you can select the directory pathname by a tree search through all directories on a volume. When applicable this is indicated on the screen, and the ? key starts the process. If you type the ? as first character then disk devices will be scanned and the volumes on line will be displayed for selection for the tree search. If you type, or move the cursor over, the directory name and THEN type ?, the device search will not be done and the tree search will be done on that directory. Utilities maintains two default pathnames. These can be switched by pressing ^X at the first character of the first default name.

In addition to the general line editing features discussed before, input of directory pathnames in Utilities has these features: Holding down the Apple key together with a number key from 1 to 7 will substitute the corresponding PREFIX for the default path given. When indicated, and when the cursor is on the first character during directory name input, pressing the Option key will bring up a request for the device number. The specified device will then be accessed through the tree search. As is true throughout ProSel-16, pressing ? at the device prompt allows you to select from a list of devices.

COPY FILES

This is a batch file copier. It retains both create dates and modification dates of files. It is compatible with all types of files including forked files, sparse files and subdirectory files. In the case of subdirectory files, however, you will be asked if you want the files INSIDE to be copied - otherwise the new directory is CREATED rather than copied, if it does not already exist. If "prompting" is OFF then the answer to this question will be taken automatically as "Yes".

You specify the source directory and the destination directory. Then a list of the files in the source directory is displayed. Move with the cursor keys to highlight the files you want copied and select them (or deselect them) by pressing the space bar. When ready to copy, press RETURN. If you decide not to copy any files press ESCAPE.

You can select/deselect ALL files by pressing ^A. (Actually, this "toggles" the flags so that previous selections will be deselected and vice-versa.) If "prompting" on the Utilities menu is OFF then files will be copied whether or not they exist or are locked on the destination directory. If prompting is ON then you will be prompted if a file of the same name exists on the destination directory, and you will be prompted again if it is locked. During prompting the file name in question will blink. Files that are being copied or that have been copied will show in inverse. When the copying is completed you will be asked if you want to copy the same files to another disk with the same directory name. This allows you to back up files to several disks without going through the process of selection over and over. Also, if you press DELETE at this point you will be sent to the delete routine at the point of file selection with the same files selected. If you then press RETURN, the original files will be deleted. (Note: This question is not asked if files inside subdirectories have been copied, because needed data is no longer in memory.)

When files have been selected, you can type ^C (for "changed" files), and the program will then automatically copy only those files among the selected files whose modification date/time is later than that of the same files on the destination directory, and all files that do not exist on the destination directory. The copying will start immediately without need of pressing the RTN key. Note that if you do not select any files before using ^C then none will be copied. To copy ALL changed files, use ^A then ^C. A ^E (exists) command works the same as ^C but copies only the modified files that already exist on the destination directory. The ^B key is similar but copies only those files whose "backup bit" is set, and clears that bit. The

DEL key works in a similar manner but will mark for DELETION from the source directory all selected files that exist on the "destination" directory with at least as late a revision date. (Note that although this is in the COPY routine, this use of DEL does not copy anything.) In contrast to the ^C, ^E and ^B commands, DEL does not take immediate action and leaves you in select mode, but remember this is now selection for DELETION.

TYPE FILES

This option will type text files (or any files) to the screen. You can select any number of files for display. The catalog information for the file is shown just previous to the file display. Most keys will stop the display or restart it. The ESC key aborts the display and returns to the utilities menu. The RETURN key aborts listing for the current file and proceeds to the next selection if any. At the end of each file display, the program will stop for a key press before continuing with the next file, or to the utilities menu in the case of the last file. The S key slows the display.

LOCK, UNLOCK, HIDE, UNHIDE or DELETE FILES

These operate much as does COPY FILES, but only those files that are appropriate to the operation are displayed. (Eg., for LOCK, only the unlocked files are displayed.) The delete function can delete entire directories, but you are warned first. You get only one warning even though you may have selected several directories for deletion. Unhide will only work if the parameter allowing hidden files to be seen in Utilities is set (see Modify parms).

RENAME A FILE or VOLUME NAME

After selection, file names are displayed at the bottom of the screen and can be changed from the keyboard. In order to change the VOLUME name, just select some file in the volume directory for renaming and, when presented with that file, move the cursor back and change the volume name. Press RTN when the cursor is at the end of the changed volume name. This will change the volume name and then present the file again with the new volume name. Just press ESC if you don't want to change the file name.

EXHUME FILES

This function lets you revive deleted files, provided the file has not been overwritten. If a problem is detected in this attempt then you will get a "file damaged" error message.
It might be wise to use Volume repair to check for any problems if anything strange happens with this function. An exhume done on the boot volume will probably cause a forced reboot to avoid

serious problems that would otherwise occur.

VERIFY FILES

This will read specified files to test for bad blocks. If an error occurs then the block will be shown at the bottom of the screen and the program will wait for a keypress. The RTN key will continue from this point, DEL will continue from the next selected file, and ESC will abort. Some disk errors on blocks not really inside the file (finding the file for example) may cause errors aborting the routine. Directory files are not themselves verified, but the files INSIDE directories will be verified if you ask for that to be done (automatic if prompts are off). [You could use Zap to read through a DIR file if you are having trouble with one.]

DUMP FILES

This will do a hex and ascii dump of files (mainly of use to programmers). It allows two way scrolling through the file. For technical reasons this will access only the first 64K of the file. (The DUMP shell command will dump the entire file but does not have two way scrolling.)

SORT DIRECTORY

This powerful directory sorter asks for a directory name and then displays the names in that directory. Some instructions appear at the bottom of the screen. It accepts the following commands:

A - sort alphabetically
C - sort by creation date
M - sort by modification date
T - sort by file type
P - sort by file type and alphabetically within a type
R - reverse present file order

You can also use the Apple key in conjunction with a cursor key to move file names around by hand.

When you are done, press RETURN. (Or press ESCAPE to abort.) When you press RETURN you will still be given a chance to abort before the sorted directory is written to disk.

FORMAT OR ERASE A VOLUME

These will erase all files on the designated volume. It gives you a chance to change your mind before the action is taken. ERASE is like FORMAT but is much faster and assumes the disk has been previously formatted.

CHANGE FILE DATE

This routine allows you to change the modification and creation dates on any file - even the volume date (which has a creation date only). You specify the pathname to operate on and you will be shown the existing dates and allowed to modify them. Just press RETURN to accept the date shown. When you are done you will be given a chance to abort the routine before the new data is written to disk. The main use for this routine is to allow you to make meaningful creation dates for files having no dates or ones on which the date was ruined by use of dumb utilities like the FILER. You do not have to type the dashes or colon shown in the date, any non-numeric character (such as a space) will do, but you must type the data in the correct position on the screen.

CREATE DIRECTORY

This allows you to create new subdirectories. It will even create multiple subdirectories. For example, if you tell it to make a directory called /HARD1/DIR1/SUB3 and DIR1 does not exist on /HARD1, then DIR1 will be created and a subdirectory SUB3 created inside it.

SHOW FILES

This catalogs a directory. Use the up/down arrow keys to scroll forwards and backwards. At the top of the screen is shown the number of blocks used by files in the directory EXCLUSIVE of subdirectories, the number of files in the directory and the number of free blocks on the volume. If showing invisible files is enabled, they are marked with an open apple if unlocked and a closed apple if locked.

SHOW VOLUME NAMES

This looks at all mounted disk devices and shows the device number and name, volume name, number of free, used, and total blocks, and the creation date of the volume.

TOGGLE BELL

If this is OFF then the bell that is heard at some prompts will be defeated.

TOGGLE PROMPTING

This toggles the prompting state for the FILE COPY and VERIFY, etc. The current state is shown on the menu. If prompts are OFF then deleting of locked files and copying over existing and locked files is done without asking whether you are sure. In addition, copying and verifying files inside selected directories is done automatically when prompting is OFF.

COMPARE DIRECTORIES

This will compare the contents of two directories (not the contents of the files in the directories). A file listed on only one of the two columns is in that directory but not in the other. Files in both directories but of different file types are shown with their types, and those with different modification dates are shown with their dates.

COMPARE FILES (K command)

This will ask for the full pathnames of two FILES you wish compared. It then compares them and displays any differences found. A very useful feature is that it allows the adjustment of the position pointer of one of the two files independently of the other. After showing a difference (which is shown by exhibiting hex and ascii dumps of 128 bytes from each file with the differences shown in inverse), you can scroll through the files with the cursor keys or go to the next difference with the RETURN key or abort with the ESC key. For technical reasons, only the first 64K of the files will be compared, and any differences after that point will not be detected.

MOVE FILES

This will move files (put them on the destination and remove them from the source) on any given volume. This is very fast and can move whole directories. It does not write new files, but just moves the pointers to the files to the new directory.

LOCATE STRING

This is for finding a text string in a group of files. (This feature is also in the File Finder module, but works somewhat differently.) You specify the directory to check and the string to look for. Found strings are displayed in context much like the same function in the File Finder.

VOLUME COPY

This volume copy facility will use as much memory as it can find in order to copy in the most efficient manner. Its operation is fairly self-explanatory, but there are two provisions that are not announced on the screen and which are for special situations:

(1) If the Apple AND the Option keys are held down when you press a number key in answer to the "Destination device" question, then ALL blocks will be copied. Normally the routine copies only USED blocks for more speed.

(2) If the Apple AND the Option keys are held down when you press a number key in answer to the "Source device" question, then the size of the disks will be assumed to be 1600 blocks (3.5" disks) and the program will ignore any unreadability of blocks. This feature is intended for dealing with badly damaged disks.

<<< Using a mouse with Utilities >>>

A mouse can be used to select options and files. It works a little differently from most mouse interfaces since it "wraps" around the screen. (Thus, for example, if the cursor is at the top of the screen then an upward mouse movement will move the cursor to the bottom, as does an up arrow.)

You can use the mouse button to select options from the utilities menu or to select or de-select files when a list of files shows on the screen.

When you are asked for a pathname, the mouse button is interpreted as a "?" (when that is accepted). The mouse can be used to move the cursor across a pathname. Thus you could, for example, use the mouse to click on "Catalog" on the utilities menu, then click again when asked for a pathname. This will bring up a list of all volumes on line. Clicking on one will produce the directory tree, and positioning the cursor and clicking on the tree will produce the desired catalog of that directory. Move the mouse up and down to scroll the catalog if it is large enough. A final click, anywhere, returns you to the utilities menu.

In the tree structured directory display, the mouse button is interpreted as a RTN to select the currently displayed directory, and mouse movement is accepted in place of cursor keys.

In a sort, the mouse button can be used in place of the Apple key to move, with the mouse, files around the screen.

In a catalog, you can use the mouse to scroll the catalog (as the up/down arrows do). The mouse button returns to the utilities menu.

GETTING RID OF PROBLEM FILES

Sometimes some glitch in the system will ruin a file to the extent that it cannot be deleted by ordinary means. There is a "secret" (meaning not shown on the screen) provision in Utilities that will allow you to get rid of such files. It is in the SORT routine, even though it has little to do with sorting. If you highlight the LAST file in the list and press Apple-DELETE then that file will be deleted from the list. Nothing happens on disk until you press return and ask for the "sorted" directory to be written to disk. To delete a file which is not the last one on the list you must move that file to the end, using the arrow keys with the Apple key, and then press Apple-DELETE.

This only gets rid of the file as far as the directory is concerned and does not free the blocks used by the file and release them to the system. To do that you should then use that provision in Volume repair.

Please note that files deleted in this fashion are not recoverable. You must use it with extreme caution. Entire directories can be deleted this way, by using it on the name of the directory.

For technical reasons this method does not work on a file that is the only file in its directory. However, the directory containing it can be deleted.

Screens:

ProSel-16 also supports extra "screens". There is a demo of these in the /PROSEL/PROSEL.16/SCREENS directory, and that demo includes some documentation on their usage. Just press 4 on the keypad and select the item SCREEN.DEMO.

ProSel-16 screens are identical to the screens in ProSel-8 as far as their outward workings is concerned, but they differ in file structure and therefore have a different file type. ProSel-8 screens can be imported into ProSel-16 by loading them into the ProSel-16 editor and resaving them (by the S command) to somewhere else on the disk.

A "screen" is just a ProSel-16 application list that is saved by the editor into a special type of EXE file that can executed by ProSel-16. It is in the nature of a screen, however, that it leaves ProSel-16 active and just replaces the application list by the one in the screen. This provides a very flexible extension of ProSel-16 which is virtually unlimited since there is no limit on the number of screens that can be used nor on the location of these files on the disk.

There is really nothing much to know about screens. They can be loaded into the ProSel-16 editor, edited and saved back again. There is nothing different about this from editing the standard ProSel-16 listing, except that you use the "Save screen" function in the editor rather than the "Save new PROSEL.SPECS file" function. Screens are "selected" just like any other executable program. It is recommended that screens be kept in the prefix #4 directory which defaults to */PROSEL.16/SCREENS if not changed in the parms. That way they can easily be selected by pressing 4 on the keypad and then selecting the desired screen file.

Although there is nothing different in screens than in a regular ProSel-16 listing, the demo screens show how one can make a screen look quite different from a standard application list. (The same things can be done with ANY ProSel-16 listing, however.)

There is a command (a ^B anywhere in any application specification screen title) that will cause the usual bottom of the ProSel-16 screen to be hidden from view and enabling regular titles going all the way to the bottom of the screen. To put titles below the usual bottom boundary, you use the ProSel-16 editor, place the cursor on the lowest line it will go to, and hold the Option key down while pressing the down arrow. This will "release the lock" and permit the cursor to go down to any line below the usual bottom.

There is another command (^C in any title) that disables the usual ESC and number key usage in ProSel. This is so you can force selection of some item on the screen and disallow exit through the "back door".

For further ideas about how "screens" can be used, go through the screen demo. Simply press 4 on the KEYPAD and then select the item called SCREEN.DEMO. That, in turn will allow you to access further screens which talk about this feature. Also, to see how some things are done, use the editor to look at the actual application specifications in any of the example screens. It is not recommended that you try this feature until you are thoroughly familiar with other aspects of ProSel-16.

Display features in a screen title:

You can use inverse text and Mousetext in screen titles as follows.

First note that to insert a control character that otherwise would be taken as a command by the editor, you type the ^O key (this means the control-O key and not the two keys ^ and O), and then the control character you want to insert.

To make an entire title show up in inverse text, just put in the ^@ character at the beginning of the title.

To turn on inverse for just a portion of a title, insert a ^O at the point you want inverse to start, and ^N where it should end.

To use mousetext in a screen title, turn mousetext on by inserting the character ^[(this is the ESC key). To turn it back off use ^X. [Note: The usual signal to turn mousetext on is the sequence ^[^O, and to turn it off is ^X^N, and these sequences must be used in ProSel-8, but ProSel-16 accepts the simpler commands.]

Mousetext characters are given by the regular keys @, A, B, C, ..., Z, [, \,], ^, and _. Note these are uppercase, and that lowercase characters do not work when in mousetext mode.

Here is a brief summary of the command characters that tell ProSel (16 or 8) to take special actions when showing the main screen.

^B: If this is in any screen title, then ProSel will not display the text usually at the bottom of the screen, and regular application titles can be put there.

^A^N: This sequence in a screen title defeats inverse that is normally shown when the cursor is on that item. Another ^A turns it back on for the remainder of that title.

^C: This character in any screen title defeats use of the number keys and ESC key while that screen is showing.

If you use these things to make fancy titles, just remember that for a title to be "active" it must start with an letter (ascii at least $40). To make the letter invisible, follow it by a backspace (type ^O^H in the editor) and then whatever you want seen.

Switching between ProSel-16 and ProSel-8:

To switch from ProSel-16 to ProSel-8, so that P8 applications will return to ProSel-8 and not ProSel-16, just execute the PROSEL.SYSTEM file. For example, this specification does that:

```
                Screen title:   ProSel-8
                Prefix:         ?
                Application:    PROSEL.SYSTEM
                Startup:        <empty>
```

You can also go directly to a P8 application from ProSel-16 with return to P8 by using the same specification but putting the pathname of the application in the startup part of the specs. If the application is run from a directory listing (after pressing a number key) then holding SHIFT down for a moment while RETURN is pressed, will have the same effect.

To switch from ProSel-8 to ProSel-16, just execute the SYSTEM/START program (assuming ProSel-16 was installed the recommended way). For example, this ProSel-8 specification does that:

```
                Screen title:   ProSel-16
                Prefix:         ?
                Application:    SYSTEM/START
                Startup:        <empty>
```

[Installation puts both of these examples in the default listing.]

The command line processor (Shell):

The old fashioned user interface on computers was what is called a command line processor or a "shell". This is a user-hostile environment. Usually it presents you with a blank screen, a prompt, and a cursor, daring you to do something intelligent. Many old-timers, mostly programmers, like this because THEY had to learn all the commands and spent a lot of time doing it, so why shouldn't YOU?

The most familiar command line processor on an Apple II is Applesoft. If you enter Applesoft, without running a BASIC program, then you are presented a "prompt" (the] character) and a cursor (the flashing box), and Applesoft then waits for you to type a command like RUN or CATALOG, etc. Applesoft is also the most powerful such interface you are likely to see. Another, older, example is the "monitor program" which presents you with a prompt (the * character) and a cursor, and has another set of commands altogether, with which many fewer people are comfortable.

There are other, more recent, command line processors for the Apple, but they all share, to greater or lesser extent, the characteristic of user hostility: to use them effectively, or at all, you must learn the commands and sometimes some very curious command syntax. Some command processors try to ease the hostility by having one command that shows you all the other commands. So then the only thing you MUST know is that one command. Still, they do not meet present day standards of friendliness.

So, if command line processors are so passe, why does ProSel-16 have one? The main reason for this is that there are some situations that could happen, rarely we hope, that leaves ProSel-16 no way to display an application screen. For example, the file PROSEL.SPECS containing the screen data may be damaged. Since ProSel-16 is the boot program, such a circumstance leaves you in a difficult position. In such instances, however, ProSel-16 gives you the option of entering the command line processor to try to rectify the situation. (For example, you can enter the Utilities module from the command line processor by typing UTIL and then you could copy the PROSEL.SPECS file from another disk, or you could rename (directly from the command line if you want) the SYSTEM/START file and reboot into another boot program.

Of course, another reason for including such a feature is that there are SOME people who LIKE such an interface, and some people who don't like it may decide it is useful when they learn more about it.

The command line processor in ProSel-16 is basically a set of a few internal commands, and includes a provision for adding commands ("transient commands"). Some of the commands are things more powerfully done by the Utilities and other modules, but which might occasionally be more convenient to do from the command processor which is always in memory.

There is also a provision for operation by a "script", see below.

Learning to use the command line processor is in no way indispensible for using ProSel-16. You can ignore it completely if you wish.

The command line processor, like other main menu selections can be accessed by pressing ESC to enter the menu and then S for Shell, or equivalently by pressing OA-S <RTN> from the screen. There is also a convenience key CLEAR that sends you directly from the main screen to the command line processor.

Entering the processor yields a brief message indicating that you can exit back to the screen by pressing ESC or typing BYE <RTN>. The processor's prompt is the colon. [Note, for example, that if you type HELP, it will print a prompt ">". Since this is NOT the processor prompt, you are still in the Help module. Pressing RTN at this point brings the ":" prompt indicating that you are again in the command processor.]

The processor contains a number of internal commands, and external "transient" commands, which are just EXE type files contained in the prefix #6 directory.

> Any executable file in the prefix #6 directory can be executed by just typing its name from the command prompt. If the file is an EXE file then ProSel does not "shut down" and it stays in control. That is the main distinction to other types of executable files. There are other differences, mainly that

ProSel-16 passes certain "pointers" to EXE type files.

Typing HELP COMMANDS <RTN> brings a list of the built in commands and some comments on each.

Typing COMMANDS <RTN> brings a list of external commands (provided the COMMANDS program is in the prefix #6 directory).

Internal commands:

HELP <alone>	displays help topics available.
HELP <topic>	reads and displays help info on <topic>.
?	short for HELP.
DELETE <filename>	deletes <filename>.
UNLOCK <filename>	unlocks <filename> for read, write, delete and rename.
LOCK <filename>	locks <filename> for write, delete and rename.
RENAME <file1> <file2>	renames <file1> to <file2>.
PREFIX <directory>	sets prefix 0 to <directory>.
PREFIX #	sets prefix 0 to prefix # (1 through 31).
PREFIX #=<directory>	sets prefix # to <directory>.
PREFIX <alone>	prints a list of prefixes 0 through 31.
PFX	short for PREFIX.
REFRESH	reloads PROSEL.SPECS file from prefix #0 volume. (Scans devices if not found.)
POP	removes one level from prefix 0.
CAT or CATALOG	catalogs current prefix directory.
CAT <directory name>	catalogs named directory.
PRINTER	turns printer on.
CONSOLE	turns printer off.
ED	goes to ProSel-16 editor.
UTIL	goes to ProSel-16 utilities.
BYE	quits command line mode and goes to ProSel screen.
QUIT	quits ProSel with no return, same as Quit menu option.
BRUN <pathname>	runs an executable file (S16/EXE/SYS type).
RUN <pathname>	runs a BAS program or BIN program under BASIC.SYSTEM.
- <pathname>	runs any program of type S16, EXE, SYS, BAS, BIN or TXT (script file).
- <path> <startup>	runs <path> (S16/SYS) giving it the startup <startup>.
CREATE <pathname>	creates DIRECTORY <pathname>.
COPY <path1> <path2>	copies file <path1> to file <path2>.
DEVICE	lists devices for choice & sets prefix.
DEVICE <1 - 9>	sets prefix 0 to volume in given device #.
EJECT	ejects device of current prefix 0.
EJECT 0	ejects all ejectable devices.
EJECT <1 - 9>	ejects disk in device # given.

BACKUP <startup>	goes to Backup passing <startup>.
REPAIR <startup>	goes to Volume repair passing <startup>.
HIDE <filename>	makes <filename> invisible.
UNHIDE <filename>	makes <filename> visible.
EXEC <textfile>	executes shell commands in textfile.
HOME	clears screen.
ECHO <text>	prints <text> to screen.
WAIT :hr:min:sec	waits until given hour:minute:second.
WAIT >hr:min:sec	waits for time period given by hr:min:sec.
RECORD	begins recording session.
RECORD <pathname>	ends recording session, writing new EXEC file.
APPEND <pathname>	ends recording session, appending to EXEC file.
PICSHOW	runs the picture show.

Separators, such as the space between <path> and <startup> can also be commas or some other characters. The < and > characters are not typed, and are shown here just for clarity.

Anything you type that cannot be recognized as an internal command is taken as an external command, meaning that the first word of the command is searched for as a filename in the prefix #6 directory, and that file is executed if found. If there is no such file then a syntax error is reported. The external commands, of course, can be supplemented or reduced by adding or deleting files in the prefix #6 directory. The ones presently there are:

ASCII	prints ascii/mousetext table.
BOOT5	boots disk in PHYSICAL slot 5.
COMMANDS	prints list of INTERNAL commands.
PURGE.MEM	purges purgable blocks from memory showing results.
WHERE	shows all memory handles and their owners.
TYPE pathname	types the TXT file specified.
DUMP pathname	does hex dump of file specified.
PATH filename	searches for filename and sets prefix #0 to it if found.
FILETYPE pathname,TYP	changes filetype of specified file to TYP.
AUXTYPE pathname,$1234	changes auxtype of specified file to $1234 or whatever.

The first five commands, that don't need parameters such as filenames, can also be executed from the main screen by pressing key 6 on the KEYPAD and then selecting the desired command file.

In FILETYPE, TYP can be in hex as in $12, or ascii as in BIN.

Beginners, or anyone in doubt about the consequences, SHOULD NOT USE the FILETYPE and AUXTYPE commands.

Other features of the command processor:

The command processor has some convenience features. One of these is the automatic expansion of commands. This is active if you use the ENTER key on the keypad to terminate the command instead of the RETURN key. [Note: the modifiable parameters include an option to interchange the action of these two keys, to defeat it altogether or to support it on both keys.] Thus, for example, if you type C SYSTEM <ENTER> the command will be expanded to CATALOG SYSTEM and the directory SYSTEM (if there is one off the present prefix) will be cataloged.

Also there is a history provision, in that the command processor remembers your last 15 commands. You can get the last command you issued by pressing the up arrow key. This prints the last command and places the cursor at the end of it. You can just press RETURN or ENTER to issue the command again, or you can edit the command a little and then press RTN. Another up arrow gets the next to last command, etc., and the down arrow moves through the list in the opposite order.

Automatic operation and scheduling:

The command processor's EXEC command reads a named text file (called a "script") and acts on its contents, assuming it is a set of shell commands. This can be used to run a sequence of programs automatically. For example, suppose you want to run the program MYPROG in the MISC subdirectory with a startup specification of STUFF, and then run the program BASIC.PROG, a BAS file, in the BASIC subdirectory. Then make a text file, say TF, with these commands:

```
   HOME
   ECHO Running MyProg with Startup STUFF.
   PREFIX */MISC
   -MYPROG STUFF
   HOME
   ECHO Running Basic.Prog.
   PREFIX */BASIC
   RUN BASIC.PROG
```

Then these commands will be sequentially executed when you type

```
   EXEC TF
```

in the command interpreter, assuming TF is in the default directory.

Any text editor that will put a carriage return at the end of each line can be used to write such a text file. For example, the Merlin assembler editor can be used for this. There is also a built-in facility for creating EXEC files (see below).

You can also EXEC a text file from a screen specification by letting the "application" be the script file. No "startup" is needed. Assuming TF is in the EXEC.FILES subdirectory, this might look like:

```
                Screen title:   Exec TF
                Prefix:         ?EXEC.FILES
                Application:    TF
                Startup:        <empty>
```

The HOME (clears screen) and ECHO (prints following text to screen) commands are intended for use in script files for the EXEC command. The WAIT shell command also is intended for use in script files. It simply waits until a certain time of day. If WAIT is followed by a time (hour:minute:second) in the form :14:45:00 then the wait will terminate at 2:45:00 PM and the next item in the script file will be executed. If WAIT is followed by a time in the format >00:30:00 then WAIT will wait that amount of time (from the present time) before executing the next line of the script. During a WAIT, the RETURN key will terminate the wait and execute the next line of the script. Similarly, the ESC key will abort not only the wait but the entire remainder of the script.

A BYE command in a script will abort the script and exit to the main screen. Otherwise you are left in Shell mode when the script is done.

Recording EXEC files:

The Shell RECORD command clears the history buffer and begins a "recording session". All subsequent lines typed at the Shell prompt are remembered but not acted upon. When you type another RECORD, but now with a pathname (eg, RECORD IT), the remembered lines will be written to the named text file. NOTE: the file is created upon the ending RECORD command and must NOT already exist or an error will occur. Since the history only remembers 15 lines, that is as much as you can send to the file. At the end of the 15 lines the command RECORD will automatically appear at the command prompt, and you must add a filename to this and write the file. However, you can also change the command to APPEND and specify an existing file. The APPEND command must have a filename attached to it and the file must exist or an error occurs. APPEND is designed to add commands to a file previously written with RECORD. You can start an APPEND session with the RECORD or APPEND command, but you must end it with the APPEND <file> command.

For example, if you type the following lines at the command prompt:

```
                RECORD
                PREFIX */MYSTUFF
                CREATE /RAM5/DIR
                COPY PROGRAM1,/RAM5/DIR/PROGRAM1
                COPY PROGRAM2,/RAM5/DIR/PROGRAM2
                RECORD IT
```

then a text file called IT will be written containing the four commands between the two RECORD lines. Then, if you type EXEC IT (or -IT) at the command prompt, the Shell will carry out those four commands.

You can put a script (or any executable file) in the prefix #6 directory and execute it by just typing its name, regardless of the setting of the default prefix.

A recording session can be aborted by exiting with the ESC key, which sends you to the main screen.

Passwords and Hiding files:

The shell command HIDE <filename> sets the "invisibility" attribute of the named file. UNHIDE does the opposite. When this attribute is set, the file will not show in a catalog in ProSel-16. (It may show in other programs depending on whether the program supports this new feature of GS/OS.) The command processor, Find File, Info Desk and Utilities all support this as does the number key feature of the selector screen. One can change that for the Shell, Find File, Info Desk and Utilities via Modify parms.

If a directory file is hidden, then all of its contents are hidden as well. If you want to hide many files or directories, consider doing it via an EXEC file, and similarly for unhiding them.

If the ProSel-16 file (usually START) is hidden, then some further restrictions are imposed by ProSel-16. In that case, access to the main menu and the shell is denied and other hidden files cannot be launched. There is only one way to override this (other than using some other program to unhide START) and that is to press CLEAR from the screen. Ordinarily that will send one to the command processor, but if START is hidden it will bring up a prompt for a password. Typing the correct password will defeat this provision for the rest of the time that you are in ProSel-16 (eg, until a reboot). Thus, at that time, you can UNHIDE START. When you HIDE START, this provision does not take effect until you exit ProSel-16 and return, for example by a shutdown. Thus you can use cycler (X from the menu) to bring up another screen and then do a warm shutdown and reboot. After that point, the user will be locked out of the menu, etc.

The password provision is not supported (and so the HIDE START will have no effect) until you provide a password. This is done via the Modify parms function. The password is case sensitive and supports any characters which Modify parms will accept as input. The length of a password is restricted to 12 characters or less.

Hiding of files can also be done from Utilities, and Unhiding can similarly be done there provided display of hidden files is enabled for Utilities (in Modify parms).

Updating ProSel-16:

Updates of ProSel-16 will be $10 if ordered from me. You must return your ORIGINAL disk with your update order. Updates will also be available (free) on the information services: Compuserve, GEnie, and AppleLink. In the latter case, the files will be encrypted and the update procedure will REQUIRE your ORIGINAL disk (not a copy). Thus keep your original in a safe place.

Quick reference:

When selecting from the screen:

Main keyboard number key X:	accesses device X.
Keypad number key X:	accesses prefix X.
Alpha key:	moves cursor to item starting with key.
OA-alpha key:	goes to that item on main menu.
CLEAR:	goes to command line processor.
ESC:	goes to main menu.

When executing a program:

Control down:	defeats "startup" specification.
Shift down:	shift to ProSel-8 if SYS & no startup.
Shift & Control down:	shift to ProSel-8 if SYS file.
Apple key down:	purge memory.
Option key down:	purge mem & force reload of ProSel-16.
Apple key & Option down:	purge, force reload, don't return.

When booting:

Control down:	defeat "boot program" in parms.
Shift down:	force running boot program (defeat day check).

Line edit keys:

^D deletes character under the cursor.
DELETE deletes the character to the left of the cursor.
^E toggles insert mode.
OA-key inserts character whether or not in insert mode.
TAB moves just past next / , or goes to line end adding /.
OA-TAB moves back to right of previous / or to line start.
^N moves to line end.
^B moves to line beginning.
^Y clears line from cursor forward.
CLEAR clears line cancelling all previous input.
ESC aborts.