CHAPTER IV

TPM-II FILE-ORIENTED COMMANDS AND UTILITIES

The following chapter will be devoted to a discussion of the TPM-II commands and utilities that apply to individual files. These commands allow you to display the contents of a file, change its name, erase the file, or change the protection level.

SYNTAX

In the previous chapter we took a peek at two TPM-II commands, DIR and LIST. Let's take a moment to expand on the proper command and utility syntax, since those covered in this chapter, as well as the next two chapters, use a common syntax.

Commands and utilities must be entered from the system prompt (A> or B>). This may seem an obvious point, but if you're used to Valdocs, there might be a temptation to try and get a directory of a diskette, for instance, in the middle of a program or utility.

In order to start a command or utility, simply enter its name, and then press RETURN. If you want to run a utility that's not on the current drive, you can precede the utility name with a drive label. For example, if the current drive is A: and you need to run LIST off of drive B: (because you don't have the file LIST.SYS on drive A:), you enter

```
A>B:LIST
```

The resident commands won't need the optional drive label because they are always in memory and available. We will identify each command and utility as such when we discuss it in detail.

The final point you'll need to learn about syntax concerns the optional command line. The commands and utilities are quite flexible, and with the proper prompting from you, they'll produce all kinds of variations on the basic task. A command line can be up to 127 characters in length, and can contain any character. We'll give you all of the possible command lines, and their effects, as we go through each command and utility.

In general, a command line is entered after the command or utility name, and before you press the RETURN key. A single space must separate them. In Chapter 3 we presented the example of selecting only certain files to be displayed by the LIST utility. By using wildcards, we were able to instruct LIST to display only the files with the filetype .TXT. Different commands and utilities will expect different command lines, but all are entered in essentially the same way.

THE DIRECTORY COMMAND (DIR)

Perhaps the most basic command is the directory, or DIR. This command is used to display the names of all of the files on a diskette. DIR is a resident command, so it's available at all times. In its most basic form, it will display all of the files in the current User Group Number on the current diskette. When you first boot TPM-II, you will be on drive A: in User Group 0. If you type

```
A>DIR
```

the files on drive A:, User Group 0, will be displayed in four columns as shown below:

```
                           **** START CRT ****

INITIAL  INI <0> | @        SYS <5> | BAUDSET  SYS <5> | COMPARE SYS <5>
ZDDT     SYS <5> | DSTAT    SYS <5> | DUMPER   SYS <5> | FILES   SYS <5>
FORMAT   SYS <5> | IOMOD    SYS <5> | LIST     SYS <5> | MODEM   SYS <5>
PIP      SYS <5> | PRINT    SYS <5> | SET-TIME SYS <5> | SPOOL   SYS <5>
SYSGEN   SYS <5>

                            **** END CRT ****
```

The files are displayed with the filenames and filetypes in columns, instead of separated by a period. The number in brackets after each file name indicates the protection level. The files are not displayed in any particular order (for example by filetype) as they are in some of the other utilities. Rather, they are displayed in the order they are stored on the diskette.

It's possible to select only certain files for display. Let's say you are only interested in the BASIC language program files (files having a filetype .BAS) on a certain diskette. By the intelligent use of wildcards, you can display only those files. To accomplish this, enter the wildcard specification in the command line. The files which match the wildcard specification will be displayed in the same format as the previous example (i.e., four columns with protection levels). To obtain a directory of only those files with the filetype .BAS, enter

```
A>DIR *.BAS
```

Obtaining A Directory Of Files In A Different User Group

If you make use of the different User Groups available on each diskette, you may like to use the "mega-wildcard" in the command line. During normal operation, DIR will only report on files in the current User Group. However, if you precede the usual command line with a User Group number, TPM-II will display the files in that Group rather than the current one. For example,

```
A>DIR 5:
```

will ask DIR to show you all of the files in User Group 5:. You can be more specific and add a wildcard specification, such as the *.BAS used above, to the User Group number. Combining the previous two examples will yield

```
A>DIR 5:*.BAS
```

To display the names of all files on a diskette, regardless of User Group, you'd enter

```
A>DIR *:*.*
```

You'll notice that the listing of the files doesn't include the User Group number of each file. DIR can tell you if a file is on the diskette, but it won't tell you which User Group number it's in. LIST will perform this task, so we'll proceed to that utility.

THE LIST UTILITY

LIST provides the same functions as DIR, along with some enhancements of its own. LIST is stored on a diskette as LIST.SYS. (TPM-II utilities all have .SYS as filetypes.) Before you try to use LIST, make sure it is on the diskette.

LIST displays the files on the diskette in alphabetical order. First, files are sorted by filetype and then by filename. The filetype is shown in the first column, and files with that filetype are displayed, three per line. After the filename, several statistics about the file are shown. The User Number appears first (followed by a colon), then the Protect Level (in brackets), and finally the size of the file. File size is measured in *kilobytes*, which is 1024 bytes or characters (a kilobyte is often abbreviated *Kbytes* or simply *K*, as it appears in the LIST display).

In addition to the data on each file, LIST displays the current time, date, and drive label before the files are listed. At the end of the file list, the current number of directory entries is shown. A small portion of each diskette is reserved as a directory area. The directory contains the name of each file and information about where that file is stored on the diskette. Each diskette has room for 128 directory entries. Some files are so large that they need more than one directory entry to keep track of all the data contained in them, and that's why there are two number listed at the bottom. The first number indicates the number of files on the diskette, while the second (in brackets) is the number of directory entries. If all of the files are sufficiently small, that they only require one directory entry, these numbers will be the same. If, however, certain files require more than one directory entry, the second number will be larger than the first. In the example below, one of the files (BIGFILE.TXT) requires two directory entries, so while there are 18 files on the diskette, 19 directory entries are used up:

**** BEGIN CRT ****

```
    Drive A:        06/28/83        15:32:11

INI=> INITIAL     0:<5>  2K
SYS=> @           0:<0>  2K | BAUDSET     0:<0>  2K | COMPARE   0:<0>  2K
      ZDDT        0:<5>  4K | DSTAT       0:<5>  2K | DUMPER    0:<5>  2K
      FILES       0:<5>  2K | FORMAT      0:<5>  4K | IOMOD     0:<5>  2K
      LIST        0:<5>  2K | MODEM       0:<5> 16K | PIP       0:<5>  4K
      PRINT       0:<5>  2K | SET-TIME    0:<5>  2K | SPOOL     0:<5>  2K
      SYSGEN      0:<5>  2K
TXT=> BIGFILE     0:<0> 36K

Total Directory entries = 18 [ 19]
```

**** END CRT ****

LIST will accept the same parameters in the command line as DIR. You can use any legal wildcard specification, as well as User Group number designation. One of the advantages of using LIST over DIR is that LIST displays the User Group number of each file. If you are unsure as to the exact number of a file, entering the command

```
A>LIST *:*.*
```

will display all files in all User Group numbers. This makes it easy to locate a specific file.

THE MASK COMMAND

MASK is an enhancement to the DIR command that allows you to specify which files will be displayed, based on Protection Level. Files are assigned one of seven possible Protection Levels. These levels are set using the PROT command, which is described below.

MASK allows you to define which Protection Levels will be displayed in one of four ways:

Less Than Or Equal To: All files which have a Protection Level less than or equal to a number 0 through 7. When you boot TPM-II the Mask is set to "less than or equal to Protection Level 5" This is denoted "<=n" where n is the upper Protection Level you wish to display.

Equal To Or Greater Than: The inverse of the above. All files with a Protection Level equal to or greater than a specified number will be displayed. Equal to or greater than is notated "=>n".

Equal To: Only the files with a specified Protection Level will be displayed. This is specified by "=n".

Not Equal To: All files except those with the specified Protection Level will be displayed. The not-equal-to sign is "<>".

The MASK command is executed by typing "MASK" followed by one of the four Protection Level specifications shown above. For example, if you want to display only files with Protection Level 5 (unerasable files, usually programs), enter the following:

```
A>MASK =5
```

Once set, a Mask will stay in effect until a new MASK command is executed, or you re-boot the QX-10. The Mask that is set using this command affects only the DIR command. The other TPM-II commands (such as ERA) are not affected, nor are the utilities, such as LIST (as one might expect).

THE PROTECT COMMAND (PROT)

The PROT command in TPM-II allows you to assign one of seven protection levels to each file on a diskette. Protection levels don't make files destruction-proof, but they do provide an additional safeguard. (Definitions of each protection level are listed in Table 3 - 1 in Chapter 3.) In most cases, you will use levels 0 and 5.

Level 0 (no protection) is best used for all data files. This will allow you to modify data without worrying that you placed the wrong protection level on your work file and that suddenly your application program will bomb out because it can't access the file. You might wish to change the level later if the file becomes an "archive" file, but if it's currently in use, it's best to leave the protection set at level 0.

Application programs and utilities are usually assigned Level 5 protection. Since these files are not modified (written to) during normal situations, this protection level prevents modification or erasure.

This is not to say that the other protection levels have no value. Once you have become familiar with your applications, you may find certain files or groups of files that lend themselves naturally to one of the other levels. If you're just starting out, though, sticking to levels 0 and 5 might be a good idea until you thoroughly understand protection levels.

We must insert a word of caution about using protection levels. Many application programs are written with the CP/M operating system in mind. CP/M doesn't support protection levels, so some application programs won't work properly if the files have a protection level higher than 0.

The protection level of a file is changed by entering the command PROT, followed by the name of the file you wish to change, and the new level enclosed in brackets. The following example will change the protection level of the file GENLEDGE.DAT to 3.

```
A>PROT GENLEDGE.DAT <3>
```

You can use wildcards with PROT, so that you can change the protection level of a group of files. If you wanted to assign a protection level of 5 to all of your .SYS (TPM-II utilities) files, you would enter

```
A>PROT *.SYS <5>
```

Both the DIR command and LIST utility display the protection level of each file. The number enclosed in brackets after each file name is the current protection level.

THE RENAME COMMAND (REN)

From time to time you'll want to change the name of a file. For example, let's say it's time to make an archive copy of a file (we will discuss the working, back-up, and archive files in detail in the next chapter). To do this you'll use the REN command.

The REN command syntax is fairly simple. After you enter REN, type the name you want the file changed to, and then the current name, separated by an "=" (equals sign). For example, to change OLDNAME.DAT to NEWNAME.BAK enter

```
A>REN NEWNAME.BAK=OLDNAME.DAT
```

REN will accept wildcards, so you'll be able to rename groups of files with a single command. The best way to explain how this feature works is by example. Let's take two different groups of files, Group A:

```
CURRENT.MAR    CURRENT.APR    CURRENT.MAY
```

and Group B:

```
LIST.SYS       COPY.SYS       PIP.SYS
```

Here we are in October, and data from March, April, and May isn't too current any more. You decide that OLD.MAR, OLD.APR, and OLD.MAY are much more appropriate names. Of course, you could rename each one of them individually, but using a wildcard is much easier. Entering

```
A>REN OLD.*=CURRENT.*
```

will accomplish this. When wildcards are used with the REN command, they mean "keep whatever is matched by the wildcard the same and change the rest". In this example our files in Group A would be changed to

```
OLD.MAR        OLD.APR        OLD.MAY
```

Likewise we could change our files in Group B so that they had a filetype of .COM with the REN command

```
A>REN *.COM=*.SYS
```

There is one limitation to using wildcards with the REN command: wildcards will only work if no other file has the same name or filetype as the group you are renaming. In the example just given, if the disk contained any other file with a similar filetype, i.e. .COM, the REN command would not work. Likewise, in the previous example, any files with a filename of OLD will prevent REN from executing properly.

THE ERASE COMMAND (ERA)

As you begin to use you QX-10 you'll notice a particular phenomenon: "disk clutter". Disk clutter can be as insidious as weeds in the summertime, and is as certain as death and taxes. It stems from the fact that floppy diskettes are finite in nature (376K worth to be exact), whereas your file collection grows without limit.

The ERA command will help you keep your data beast in check. Its purpose is to erase those files which are no longer needed. Many of your files will be replaced by more "current" versions, temporary files will outlive their purpose, and these files will unnecessarily occupy space on your diskettes.

ERA removes a file permanently. When a file has been erased, it will no longer show up in the disk's directory, and the space previously reserved for the file will once again become available. To erase a file, type ERA followed by the file name. For example, to erase the file OLDTIMER.FIL, enter

```
A>ERA OLDTIMER.FIL
```

In addition to erasing individual files, ERA will accept wildcards. You can erase groups of files, using the proper wildcard specification, or you can even erase all of the files on the diskette (within the current User Group) with the "*.*" wildcard.

One of the aspects of ERA that separates it from other TPM-II commands and utilities is the fact that its effects are irreversible! Therefore, use ERA with care! If you use want to use a wildcard with ERA to erase a group or groups of files, it is a good idea to examine DIR first, using the wildcard, so that you're absolutely sure of what you're erasing. To further minimize the chance for catastrophic mistakes, ERA will respond with the prompt "ALL FILES? (Y/N)" if you place wildcards in both the filename and filetype. For example, if you want to erase all files on a diskette, enter

```
A>ERA *.*
```

The ERA command will respond with

```
A>ERA *.*
ALL FILES? (Y/N)
```

Responding with anything except "Y" will result in ERA ignoring the request and returning you to the system prompt. This safety prompt will appear each time you place wildcards in both the filename and filetype, even when it doesn't specify all of the files on the diskette. The command

```
A>ERA PROG*.*
```

will illicit the prompt, even though the only files which will be erased are those that have "PROG" as the first four characters.

THE TYPE COMMAND

The final command we'll cover in this chapter is the TYPE command, which is useful for identifying the contents of certain files when it's not readily apparent from the file name. Entering the command "TYPE", followed by a filename will instruct TPM-II to display the contents of the file on the screen.

The conditional clause here is that the file must contain text, or, in some cases, data which is in "human readable" form. Try using TYPE with one of the .SYS files on your diskette. The program instructions contained in this files make perfect sense to the QX-10's CPU, but to you they'll look like a meaningless series of letter and graphics symbols darting around the screen in no apparent order.

If the file is of any length, it will probably scroll off the screen faster than you can read it (TYPE may be viewed by some as a computerized version of the Evelyn Wood Reading Dynamics course). In order to stop the display, type a CONTROL-S (hold down the key marked CTRL while you type the letter "S"). The screen will freeze until you press any other key. You can stop the display and restart it as many times as you wish.

If you have a particularly long file, and you don't want to display the entire file, you can exit the TYPE command at any time by typing CONTROL-E. The TYPE command will not accept wildcards for obvious reasons.

ONWARD AND UPWARD

In Chapter 5, we'll present the disk-oriented section of TPM-II. By now you should be fairly familiar with TPM-II commands.

END CHAPTER 4

CHAPTER V

TPM-II DISK-ORIENTED COMMANDS AND UTILITIES

In the previous chapter we covered the portion of TPM-II that deals primarily with individual files. Here we'll examine disks and groups of files, that is: how to create new work disks, moving files from one disk to another (including proper back-up procedure), and maintaining a library of disks and files.

For the most part, the commands and utilities you'll deal with in this chapter are the same as those we've encountered previously. They use the same syntax: wildcards, file names, drive labels, etc., but some of the commands are slightly broader in scope, and the tasks they execute are not quite as straightforward. Consequently, several will engage you in a dialogue once they've been invoked: you'll be asked to answer questions about the tasks you want accomplished.

WHAT IS "DISK MANAGEMENT" ANYWAY?

When you first begin using your QX-10, the concept of "disk management" might seem like a molehill masquerading as a mountain. After all, how difficult can it be to manage one disk with 10 files on it? However, if you've been using your QX-10 for very long (say under Valdocs), you've no doubt noticed that your disks keep filling up and multiplying.

The art of disk management, then, is one of conquering the data beast that dwells within your system. Its daily sustenance is the unused space on disks which it devours with a ravenous appetite. Although the beast can't be vanquished, it can be checked.

Disk management is nothing more than learning to use a few of the tools provided by TPM-II for this purpose, and developing a set of good habits. Most disk management stems from what we call the "nature of disks". The most obvious observation that comes to mind is that disks are square and have a round hole in the middle. Beyond that, they store data in the form of files, and they get filled up. This is one of the principal reasons hard disk drives are often purchased by QX-10 and other personal computer owners. When floppy disks fill up, they tend to multiply as files spill over from one disk to another. Thus, experienced QX-10 users will more likely describe a disk as "a full and plentiful object, commonly associated with computers."

One of the less obvious characteristics of a disk (hard or floppy), but equally important from a disk management point of view, is its vulnerability. From the vantage point of a floppy disk, it is a harsh world indeed. Smoke, fingerprints, magnetic fields, temperature extremes, and a host of other environmental factors make their job of holding onto data quite difficult. Hard disks are shielded from many of these factors, although they are still subject to mechanical shock, failure, and several other factors. Still, as a matter of record, hard disks fail significantly less frequently than floppy disks.

This is not to say that disk management is unnecessary with hard disk drives. Quite the contrary -- the fact that they store between 5 Mbytes (million bytes) and 40 Mbytes raises the stakes so high that you can't afford to gamble.

DISK MANAGEMENT, TPM-II, AND YOU

There are five steps to good disk management. These steps or rules will make your use of the QX-10 much easier and more rewarding by insuring that your data and programs are safe and secure.

Orderly File Names: We discussed this earlier in this manual, so we won't dwell on it here. The importance of consistent and descriptive file names can't be over-emphasized. The TPM-II file structure is flexible enough to allow you to choose file names that are somewhat descriptive.

Proper Handling Of Disks: Treat your disks with respect! They are basically rugged, but they are in no way immune to mistreatment. Pages 1 through 9 of your Valdocs manual list precautions about disk handling. Please re-read it.

Frequent Back-up Copies: You must get in the habit of making copies of your work disks frequently. Making a copy every hour is not excessive. Remember, the amount of data you could loose in the event of a power outage, computer failure, or other catastrophe is whatever has been entered into your QX-10 since you last made a copy of it. At a minimum, you should make a copy every day.

Work Disks For Different Applications: Once you are used to your QX-10, you'll find it can be used for many different tasks or applications. Each application should have its own work disk containing the application program and data files. We'll discuss creating and using work disks a little later.

Maintain A Good Physical Storage System: Ironically, many QX-10 owners maintain their files correctly only to find that their system is for naught because they can't find the correct disk when they need it. Set up a storage place where you can keep your disks in some sort of order for quick retrieval.

TPM-II will provide you with the tools (utilities) necessary to follow these guidelines for proper disk management. By the end of this chapter you should be familiar with all of the commands and utilities necessary to implement your own specific disk management system.

As you can see from these pointers, TPM-II won't do it for you. It will help, but the burden of good disk management lies with you.

SYSTEM AND DATA DISKS

As you'll recall, Valdocs uses both System and Data disks. When you run applications under TPM-II, you'll still have System disks and Data disks, but their functions will be slightly different.

Valdocs is made up of several modules or programs which are combined into what appears to be a single program. Many larger application programs are also structured this way. It's not uncommon to find a General Ledger program which is comprised of some 10 to 15 subprograms that are connected or chained together in such a way as to appear to be one large program.

In certain applications, subprograms are so extensive that an entire disk is required to store them, leaving no room for data. This is the case with Valdocs, and you will no doubt run into other applications that are similar. Such programs require that you have a system disk, on which the application program(s) and TPM-II operating system are stored, and a data disk where the data resides.

A system disk, then, is a disk which contains TPM-II and the application program(s). The system disk must contain a copy of the TPM-II nucleus in order to boot (see Chapter 3). As long as the system disk in drive A: has TPM-II, you don't have to have TPM-II on your data disk. However, as you will see shortly, omitting TPM-II from your data disks doesn't increase the amount of storage space available, so we recommend that you put TPM-II on all of your disks.

Although certain applications require a large number of subprograms, just as many applications don't. These programs will fit on a system disk along with TPM-II and still leave plenty of room for data files. Data disks are somewhat superfluous with these application programs. All you'll need to run them is a single system disk, containing the program, data, and TPM-II.

The documentation that accompanies your application program will often tell you specifically whether you need a single system disk or a system disk and a data disk. If it doesn't contain this information, you'll have to determine how much space is required for the application programs, and whether or not there will be enough space for data on the disk. If you're at all unsure, it's best to go with two disks.

CREATING NEW DISKS

The first step in starting a new application is to create a work disk or disks. As you use your QX-10 you will need to create new disks often, whether for work, back-up, or archive disks (we will go into back-up and archive disks shortly). The process we describe will be almost identical for each type, but in order to take things in a logical order, we'll start with the work disk.

How Information Is Stored On A Disk

The word "creating" is a bit misleading when we talk about disks; you are undoubtedly aware that you don't create disks, you buy them. When we say create, we really mean "put data onto." The first step in creating a new disk is to format it.

The QX-10 stores data on disks in 512-byte chunks called sectors. Ten sectors are laid out end-to-end around the disk to form a track. The tracks on a disk are a series of concentric circles, each containing 10 sectors. The disks used by the QX-10 have 40 tracks on each side.

Each block of data stored on a disk has a unique address: a track and sector number. When TPM-II is storing or retrieving information it specifies the track and sector number where the data is to be stored or retrieved.

When you purchase your disks they aren't divided into tracks and sectors. The process of laying out these tracks and sectors on a new disk is called formatting. Formatting a disk is similar to surveying land -- you mark out the boundaries of each track and sector when you format. Trying to store data on a disk that hasn't been formatted would be like trying to buy a parcel of land before it has been surveyed. How can you buy it when no one knew where the plot starts or ends?

Formatting A Disk

You can format a disk one of two ways. The first, which you're no doubt familiar with, is to use the Valdocs "COPY DISK" facility. However, this requires you to swap disks, since you must be running Valdocs in order to use the COPY DISK key. The second method it to use the FORMAT utility supplied with TPM-II.

FORMAT is one of these utilities that has its own dialogue, and to which we alluded at the beginning of this chapter. To format a disk, place a TPM-II disk with the program FORMAT.SYS on it in drive A:. Place the disk to be formatted in drive B:. Start the FORMAT utility the way you would any other by typing

```
A>FORMAT
```

BE CAREFUL USING FORMAT. It will completely erase any data previously on the disk. Make sure that the disk you're formatting is either brand new or is a previously used disk with files you don't currently need. FORMAT has the capability of formatting individual tracks instead of the entire disk. In order to use a disk, you must format the entire set of tracks (40) on both sides of the disk. Therefore, when the following dialogue is displayed on the screen, answer the question by pressing RETURN:

**** START CRT ****

```
A>FORMAT

TPM Floppy Disk Format, V3.04, for "QX-10  [5 inch 10-512 ]"
Remember, drive "A" is on the LEFT, and "B" is on the RIGHT.
Drive to be Verified/Formatted? (A,B) >B

Note: If the next question is answered with
just a RETurn key, we will default to a full
format of the entire disk.  If an individual
track or side is to be formatted, or if you wish
to VERIFY, answer with a number  (0-39).

Starting Track number  (usually 0) :<RETURN>

We will FORMAT the entire disk.  This will
DESTROY any data on the disk.  Be Sure.

Now place disk in drive B.
Hit RETURN when you are ready><RETURN>
```

**** END CRT ****

FORMAT will now start formatting the disk in drive B:. The number of each track will be displayed as it's first being formatted, then verified:

```
Formatting Side 0 -Track nn
Formatting Side 1 -Track nn
Verifying  Side 0 -Track nn
Verifying  Side 1 -Track nn
```

If any sector is defective, i.e., FORMAT can't read or write to it, an error message will be displayed indicating which sector is defective. If this happens, try reformatting the disk again. If the same thing happens, DON'T USE THE DISK. You will probably end up loosing valuable data.

When the verification is complete and no defective sectors are found, FORMAT will print the following message:

```
Disk Verification Complete
Another Disk (Y/N)
```

If you answer "N", FORMAT is done and will return you to the system prompt. "Y", on the other hand, will restart the dialogue from the beginning, thus allowing you to format a batch of disks at one sitting.

Now that the disk has been formatted, we must place a copy of the TPM-II operating system on it, along with the program and data files you want. Remember, formatting only marks the track and sector boundaries on the disk. It doesn't place any program or data files on the disk. TPM-II provides two additional utilities to accomplish these functions.

Transferring The TPM-II Operating System To Your New Disk

The SYSGEN (short for SYStem GENeration) utility is used to transfer the TPM-II operating system from one disk to another. The first two tracks of each disk are reserved for TPM-II, and these tracks are always kept separate from the rest of the area where other files are stored. SYSGEN is really a special copy program which allows the contents of these two tracks to be read from one disk and stored on a second disk in this reserved area. To start SYSGEN, place a disk with SYSGEN.SYS and TPM-II in drive A: and a formatted disk in drive B:. SYSGEN is invoked using the same method as other utilities.

```
                        **** START CRT ****

A>SYSGEN

SYSGEN [for QX-10 with TPM], Version 5.19

Get system? (A,B,C, or D - RET to skip)A
Source Disk ON A.  Type return to proceed.<RETURN>
Last address = 4900H
SYSGEN function complete.
Put system? (A,B,C, or D - RET to skip)B
Destination Disk ON B.  Type return to proceed.<RETURN>
Last Address = 4900H
SYSGEN function complete.
Do another? (Y or N)N
REBOOTING TPM.  Type return to proceed.<RETURN>

                         **** END CRT ****
```

The SYSGEN dialogue will ask you which drive is the source drive. This question simply means "which drive do I read the copy of TPM-II?" Respond with "A" for drive A:. This answer will be verified by SYSGEN by restating your selection. If it's correct, press the RETURN key. SYSGEN will read TPM-II from the disk and into memory. The address of the last memory location used is printed next: ignore it. This number is provided for the edification of programmers wishing to modify TPM-II in some manner.

Now TPM-II will ask you which disk you want TPM-II transferred to. Answer with "B" for drive B:. Again, SYSGEN will verify your answer before actually storing TPM-II on the new disk. Like FORMAT, SYSGEN gives you the opportunity to process a batch of disks at one time. If you answer "Y" to the next question, the dialogue will be repeated. An "N" will return you to the system prompt.

You can verify that the disk in B: does indeed contain TPM-II by placing the disk in drive A: and pressing the RESET button located under drive B:. You will see the familiar system sign-on message and system prompt. You don't have any files on the disk yet (you can verify this by using DIR at this point), but you are two-thirds of the way through the process of making a new work disk.

Transferring Files To The New Disk

Once your new disk is formatted and contains TPM-II, you're ready to transfer the necessary files to it. The first set of files you'll want to transfer are those commonly used TPM-II utilities, such as LIST, followed by the appropriate program and data files from the application program you want on this disk.

The PIP utility is used to transfer files.

PIP has two modes of operation, the command mode and the interactive mode. We'll begin our discussion with the interactive mode, and then return to the command mode.

The primary purpose of PIP is to move files from one disk to another. PIP has several other features, such as displaying a disk's directory and concatenating several files into one, which we won't cover here. (See the advanced section of this manual. Once you have mastered the basic operation of PIP, you may wish to read about those features, even if you aren't interested in programming under TPM-II, the main topic of the second section.)

The PIP command consists of a destination filename and a source filename. The two are separated by an equals sign "=", with the destination placed before the equal sign and the source placed after the sign. Since the primary function of PIP is to move files from one disk to another, the drive is included in the filenames.

Here's an example to illustrate the PIP command syntax. Transferring the file GOOD.DAT from drive A: to drive B: would require the command

```
B:GOOD.DAT=A:GOOD.DAT
```

Notice that the destination is listed first, followed by the source. This might seem a little backwards, so be sure you get them the right way around.

PIP will accept this command line as a perfectly legal one, but there is a shorthand way to enter the same thing. If the name of the file doesn't change between disks, you can omit the destination filename and type only the drive label. PIP will "fill in" the source filename for you. Thus,

```
B:=A:GOOD.DAT
```

is interpreted by PIP the same as

```
B:GOOD.DAT=A:GOOD.DAT
```

PIP will also recognize wildcards, which makes it easy to move large numbers of files around. For example,

```
B:=A:*.DAT
```

and

```
B:=A:*.*
```

are valid PIP commands. The first will transfer all files with the filetype .DAT to drive B: from drive A:, and the second will copy all files to B: from A:.

If a wildcard has been specified, the name of each file that is matched will be printed as it is copied, thus insuring that all the files you wanted copied have been copied. If the file name is explicit (i.e., has no wildcard), the name is not printed as the file is copied.

To enter the interactive PIP mode, you invoke PIP without any command line. PIP will print a sign-on message and then display its own prompt, the asterisk character ("*"). At this point, you can enter a command, which PIP will execute, then redisplay its prompt. You can enter as many commands as you wish in the interactive mode. To exit, simply press the RETURN key instead of entering a command. PIP will terminate operation and return you to the system prompt.

**** START CRT ****

```
A>PIP

ZPIP Version 1.60
*B:=A:*.SYS
*B:=A:*.DAT

Copying:
GOOD.DAT

*<RETURN>

A>
```

**** END CRT ****

PIP's command mode accepts all of the same commands, but it will execute only one. If you enter a command in the command line when you invoke PIP, it will execute that command then return you straight to the system prompt. Let's look at our example of the command B:=A:*.SYS. This time we'll execute it in the command mode instead of interactively:

**** START CRT ****

```
A>PIP B:=A:*.SYS


Copying:
LIST.SYS
PIP.SYS
FORMAT.SYS

A>
```

**** END CRT ****

PIP recognizes the TPM-II protection levels. One of the possible uses of this protection feature is to prevent files from inadvertently being "copied over," i.e., replaced by a file with different contents, but the same file name. If the protection level of a file is set to 6, you can prevent this from happening.

The Valdocs COPY DISK function is easier to use than PIP if you are simply making an exact copy of a disk. However, PIP is more useful when creating new work disks. With the ability to copy individual files or groups of files, you can create work disks with the necessary programs and data files for each application.

You should become familiar with PIP since it is one of the most frequently used of the TPM-II utilities. We'll now look at the need for back-up copies of your work disks, and how they're created.

CREATING BACK-UP COPIES OF YOUR DISKS

The importance of making back-up copies can't be overemphasized. The QX-10 is a well-engineered computer with an extremely high level of reliability. The utility system in the U.S. is the best in the world. Today's application programs are well-tested products, extremely forgiving of incorrect commands and user errors. Still, if you think that Murphy simply had a bad attitude, you haven't been around computers long enough.

Making a backup copy of a disk is really very simple. When you are making new work disks, make them in twos. Label one as the work disk, with a proper description of which application the disk is for, and then label the second with the same information, adding the words "BACKUP" in a conspicuous place.

Using PIP

Each work disk should contain a copy of PIP. In order to make a back-up copy, simply insert the work disk in drive A: and the back-up copy in drive B: and type

```
A>PIP B:=A:*.*
```

This will copy all of the files from drive A: to drive B:. Make sure, however, that you don't get the source and destination drive labels reversed (i.e., don't enter A>PIP A:=B:*.*) or you'll copy yesterday's work over today's.

Making backup copies in this manner is simple and takes but a few minutes. Of course, if you are only working on one or two files at a time, you could copy those specific files instead of copying all of them; this will speed up the back-up process. However, it's a good idea to copy the entire disk at least once a day in case you worked on a file and forgot about it, or your application program updated a file of which you were unaware.

Using COPY

COPY.SYS, a second utility on your TPM-II disk for making back-up copies, is identical to the TPM-II Copy routine. COPY will allow you to make identical copies of a disk, i.e., it formats, copies, and verifies each track of a disk, rather than transferring files one at a time. (Note: COPY does have a single-file copy option, but because of the menu structure, it takes more time than PIP, and it doesn't accept wildcards. (See the Valdocs manual for instructions.)

In order to start COPY, enter

```
A>COPY
```

and the COPY menu from Valdocs will appear on the screen. (Note: One of the options in the Valdocs SETUP program is to specify a 24- or 25-line screen. COPY will not properly display the menu on a 24-line screen, although the utility will still function correctly.)

Once you've called up the menu, select "COPY A DISK". Place the work disk in drive A: and the back-up disk in drive B:, per the instructions. COPY will now make a mirror image back-up, updating all files on the disk.

Pros And Cons

Both PIP and COPY have their place in the backup process. Each has a particular feature which makes it more appropriate in certain situations.

PIP gets the nod if you've only modified one or two files and don't want to copy the entire disk. COPY doesn't allow you to specify individual files: it's all or nothing. PIP's other advantage is that it consolidates files. During normal operation, individual files tend to spread out over the disk as the TPM-II disk manager allocates and de-allocates space as required. After a file has been added to, changed, and deleted from a number of times, the records on which the file is stored may not be located together.

In some applications, this "scattering" of records can slow down disk access speed. PIP will "regroup" each file as it is transferred to the new disk. If you periodically transfer a file from your work disk to the back-up disk and then from the back-up to the work disk, you can improve the access speed.

COPY's most obvious advantages occur in the case of a complete disk back-up. If you're making a complete copy of a full disk, COPY is faster than PIP. In addition, COPY will verify that the back-up copy is the same as your original work disk by reading back each track after it is copied. PIP doesn't verify.

As you become more familiar with your application(s) and your QX-10, it will be clear which utility is more appropriate in different situations. For now, make sure you know how to use both, since each will come in handy further down the road.

MANAGING THE FREE SPACE ON A DISK

The final aspect of disk management is learning to cope with the finite nature of disks; they are far from limitless in storage capacity. You must, therefore, keep an eye on the amount of space remaining on your disks.

Obviously, you are trying to avoid a full disk. Different application programs deal with this situation with varying degrees of grace. Assume, for example, that you have been merrily working away at the 1983 sales projects and completed your task a mere 10 minutes before the meeting with your boss. You go to save the projections on a disk, and then it happens..... your program sorrowfully reports that there isn't enough room.

Where did all that room go? After all, 376K is a lot of characters.

Let's digress for a moment and look at the space allocation of a typical disk. In order to make the numbers a little easier to juggle in our heads, we'll use a word-processing application, since each byte corresponds to a character. One of your disks then can hold the equivalent of just over 100 single spaced pages. For those not given to literary excess, 100+ pages probably seems like quite a bit of room. The root of the problem is that those 100+ pages are a fiction. The disk has the equivalent amount of available space, but you won't be able to store even 50 pages of text. Let's see why.

First, TPM-II reserves the first two tracks: there goes five pages. Next, you'll want to put several of the TPM-II utilities such as PIP and LIST on your disk. Another three pages out the window. Then you have your word processor program(s); a medium-sized program can easily take up 16 or so pages. Now we've used 24 of our 100 pages, leaving us with 76. Here's the clincher. Most application programs, particularly word processors and spreadsheet generators, make a back-up of a data file each time they update it. This cuts the 76 pages in half, and our 100 pages actually turns out to be 38 usable pages.

The SPACE Command (SP)

TPM-II's SPACE command will tell you how much unused space is left on a disk. Simply type

```
A>SP
```

and SPACE will tell you, in Kbytes, how much of the disk is currently unused. Entering a drive label will inform you how much space is left on a drive, other than the current one.

**** START CRT ****

```
A>SP B:

106K Left
```

**** END CRT ****

Using SPACE And LIST Together

SPACE and LIST are a matched set when it comes to managing the space on your disk. LIST will give you a list of each file, along with its size. Combining this information with that gleaned from SPACE will give you a pretty good idea of where you stand.

A good rule of thumb is to make sure that the amount of unused space is greater than the largest file size. For example, if your largest file occupies 35K, you should consider your disk "full" when SPACE informs your there is less than 40K of unused space.

WHAT DO I WHEN THE DISK IS FULL?

So you've been diligently watching the size of your files and your free disk space, and you see that the disk is full. Now what? You must move some of the files off the disk. The first candidates are any unnecessary files. Those containing data that has been superseded and has no historical value can be erased, but first be sure that they are indeed expendable: once they're gone they're gone for good.

If there's no fat to be trimmed, the next step is to determine which, if any, of the files that are not currently in use, but that you want to save for future reference, can be transferred off your work disks onto an archive disk. These files are the cancelled checks of the computer world. They should be stored in a manner that ensures they'll be around and accessible when you need them. The same guidelines that applied to naming files and grouping them into different work disks applies to archives.

If you still don't have enough room, it's time for decisive action. Your work disk must be subdivided. If you're currently using a single work disk, you can split it into two: a system disk, containing the application programs, and a data disk, with your files (like Valdocs). If you're already using two disks, see if the data files can be split into two groups.

A FINAL WORD OF ENCOURAGEMENT

We've covered quite a bit of ground in this chapter. The topic of disk management in particular is not one that can be mastered in one sitting. If some of the things we've said are still a little fuzzy, don't worry. Take your time. Work with the examples in this manual and then try a few of your own. Remember, experience with a computer is measured in the same units as piloting an airplane -- hours. No one would expect you to be able to fly a 747 after two hours, and you certainly aren't expected to master the fine points of disk management overnight!

END CHAPTER 5

CHAPTER VI

NON-DISK COMMANDS AND UTILITIES

Most TPM-II commands and utilities deal with disks and files. A few, however, deal with the QX-10's other component parts, such as the clock and printer. We'll cover these in the next few pages.

THE SET-TIME UTILITY AND TIME COMMAND

The QX-10 contains a clock circuit that keeps track of the current date and time. The clock runs independently of the rest of the circuitry in the QX-10, and has an attached battery that supplies power, even when the rest of the system has been turned off.

The Valdocs program constantly displays the correct time in the lower right-hand corner of the screen. Most of the application programs that you run on the QX-10 will not do this, so occasionally you may need to ask TPM-II for the time.

The TIME command will display the current date and time, in 24-hour format. Simply type

```
A>TIME
```

which gives you the date and time in the following format:

```
08/03/83  19:12:45
```

Occasionally you'll need to reset either the date or time (to accomodate Daylight Savings Time, etc.). The SET-TIME utility has been provided for this purpose. When you change the time, be sure to remember that the QX-10 stores the time in a 24-hour format.

To reset either the date or time, enter "SET-TIME" followed by the date and then the time in the proper format. For example, the following will set the date to July 4th, 1983, and the time to 1:45 in the afternoon. In order to help you get the seconds correct, enter a time which is 30 seconds or so from the exact time. SET-TIME will prompt you to "Press return at "nn" seconds," which will give you time to enter the command and be ready to press RETURN at exactly nn seconds.

```
              A>SET-TIME  7/4/83  13:45:50
              Press return at "50" seconds.

              A>
```

THE PRINT UTILITY

PRINT.SYS is used to print the contents of an ASCII file to the printer and/or screen. The PRINT utility is quite flexible, allowing you several options to make sure your file is printed exactly as you want it.

The general form of the PRINT command line is

```
 A>PRINT [file1],<file2>,<file3>,...,<file6> /<switches>
```

If this looks a little imposing at first, don't worry, it's not. Let's take each segment of the command line and describe it, beginning with the filename(s).

PRINT will accept up to six file names in the command line. It has the ability to "remember" the names of all of the files, and when the first file has been completely printed, it will print the second one, and so on, until all six files have been printed. This procedure of lining up more than one file to be printed is referred to as "queuing". Of course, you don't have to enter more than one file, but if you have several files you want printed at once, queuing allows you to set all of them up and then go off and do something else while it's actually printing.

The files must be entered in the order you want them printed. The file name must be separated by either a space or a comma. You can precede the file name with a drive label, if you wish to print a file that doesn't reside on the current disk. If you omit the filetype, PRINT assumes .DOC. Therefore, if you intend to use PRINT a lot, you might want to get in the habit of naming your text files with the .DOC filetype.

After the file names have been entered, several parameters or "switches," as they're sometimes called, may be entered. These control the way in which the files are actually printed. PRINT.SYS has five switches that can be used to control the output. These switches may be used in combinations, provided they aren't mutually exclusive. The PRINT switches are:

C - Print the file to the console only.
B - Print the file to both the console and printer.
P - Print the file to the printer (List device) only. This is the default if neither C, B, or P is entered.
N - Suppress file headings.
M[n] - Indent the left margin n spaces. The default is 8 spaces.

The switches may be entered in any order, provided that there are no spaces between them. They are separated from the file list by a space and a "/" character.

The "C", "B", and "P" switches determine <u>where</u> the file will be printed. Normally, it is only directed to the printer, but is can be routed to the screen, or both, by using the C and B switches, respectively. Only one of these switches may be used at a time.

The "N" switch suppresses the file heading. Normally, PRINT will print a heading on each page which contains the file name, date, time the printout was started, and page number.

In most situations, a left-hand margin of eight characters will center the text on the paper. The "M" switch allows you to change the indentation of any number of characters.

Let's summarize all of this with a few examples. Assume you're writing a book, and you've finished the first three chapters. You'd like to print them out, but you don't want the usual PRINT heading on each page. The command to accomplish this would be

```
A>PRINT CHPT1.BOK,CHPT2.BOK,CHPT3.BOK /N
```

This will print chapters 1, 2, and 3, in order, omitting the header normally printed on each page.

Now let's take a look at a report you're preparing for a meeting that convenes in 10 minutes. You want to see a trial printout, but you don't have time to print two copies. The following command line will allow you to print a copy on the screen (which will go much more quickly than on paper) with an indentation of 2, instead of 8, so that the line won't "wrap-around".

```
A>PRINT REPORT.BIG /M2C
```

THE SPOOL UTILITY

SPOOL is a first cousin to PRINT. The SPOOL utility also allows you to print a file, but it performs this task "in the background" so you can run another program or utility while the document is being printed. Unlike PRINT, you can only feed SPOOL filenames one at a time.

SPOOL uses a special "interrupt" feature of the QX-10 to perform this Dr. Jekyll and Mr. Hyde act. Valdocs uses this feature to print a file while the QX-10 is performing some other function. Under TPM-II, this feature must be "activated," which is accomplished with the Valdocs SETUP program. One of the listed options is to "enable background interrupt and font driver support". Use SETUP to enable this feature, then use SYSGEN to copy the new version of TPM-II with this feature built into your work diskette.

You're ready to start using SPOOL. To activate it, type SPOOL, followed by the name of the file you wish to print. SPOOL will display its sign-on message, and the printer will start up. You'll then see the system prompt again, which means the QX-10 is ready to execute another command. SPOOL will continue to print out the file in the background until it's complete. When the printer stops, SPOOL is ready to accept another file to print.

**** START CRT ****

```
A>SPOOL GRTAMR.NOV
QX-10 TPM spooler. Spooling Started.
A>
```

**** END CRT ****

If the file you want to print was prepared by the Valdocs editors, enter a "V" after the name of the file. This will instruct SPOOL to print the file in the Valdocs format; otherwise it will print it literally.

BATCH PROCESSING WITH TPM-II

Certain applications require that a group of TPM-II commands be executed over and over. For example, to make a new working diskette, three steps must be followed: first, you must use FORMAT to initialize the diskette; next, SYSGEN is called to transfer a copy of TPM-II onto the new diskette; and finally, PIP transfers the necessary files to the new working diskette.

The TPM-II SUBMIT utility will help ease some of this boredom by automatically executing an unlimited number of TPM-II commands. Unlike other utilities, SUBMIT is actually stored on the diskette as @.SYS. It has two modes, an interactive mode and a command line mode. In this respect, SUBMIT is similar to PIP. The interactive mode allows you to enter a series of TPM-II commands, while the command line mode executes a set of commands that have been stored as a file. Let's take a look at the interactive mode first, and once we've got the hang of that we'll move on to the command line mode.

Interactive Mode

To start SUBMIT in the interactive mode, simply type

```
A>@
```

SUBMIT will sign-on as shown below:

```
TPM Submit Utility, Version 1.xx as of MM/dd/YY

Enter commands, one line at a time.
To EXIT, type CONTROL-Z
To ABORT, type CONTROL-C
To CANCEL a line, type CONTROL-U
To RESTART (cancel all lines), type CONTROL-E

@
```

This second "@" is the SUBMIT prompt. Like the PIP command prompt ("*"), it will be displayed before each command you enter.

We can now enter the commands we want SUBMIT to execute as a batch. Let's use the example of the new working diskettes, mentioned at the beginning of this section. We'll enter the three commands (FORMAT, SYSGEN, and PIP) followed by a CONTROL-Z:

```
@FORMAT
@SYSGEN
@PIP B:=A:*.*
@^Z
```

TPM-II will execute a warm boot (the equivalent to typing a CONTROL-C) and begin executing FORMAT. One extremely important point about SUBMIT will become apparent here: SUBMIT will not supply your responses to the prompts within a program, only valid commands at the system prompt (A>). FORMAT will ask you "Which drive to format?", etc. You must answer these prompts -- you can't enter your responses into the list of commands. Likewise, with SYSGEN, you must respond to the usual prompts. However, PIP will execute without further intervention with the command line

B:=A:*.*.

Command Line Mode

While the interactive mode has its uses, the command line mode is more powerful. Commonly used groups of commands can be stored in a file; they can then be executed any time by entering the name of the file when you invoke SUBMIT. This will save you from having to enter the commands each time the group is executed.

The commands can be entered into a file using a word processor, Valdocs, or the DDT utility. Enter the commands just as you would from the console; one command per line followed by a carriage return (RETURN). The files can have any filename, but they must have a .SUB filetype.

In order to invoke SUBMIT using a file with commands, enter the normal SUBMIT character (@) followed by a space and then the filename of the command file. You won't need to enter the filetype (.SUB); SUBMIT will automatically fill that in. Let's look at an example of how SUBMIT works with a command file.

We'll assume that those commands we presented earlier (to create a new work diskette) have been stored in the file NEWDISK.SUB. In order to make a new work diskette, simply enter the following command:

```
A>@ NEWDISK
```

Each time the system prompt appears, one of the commands in the file, in order, will be automatically filled in and that utility will be invoked. You must still respond to prompts from within the utilities, as before.

It is possible to develop a library of SUBMIT command files that will help you execute some of the more commonly used command sequences. With a little preparation (making the files) you will find the SUBMIT command to be a valuable addition to your TPM-II utilities.

THE BAUDSET UTILITY

The use of certain peripherals will make it necessary to change the baud rate of the serial port. Valdocs will set the rate to either 300 or 1200 baud (depending on the modem speed you have selected with SETUP). The serial port can be set to any standard rate up to 9600 baud. The baud rate is stored in the battery-backed CMOS RAM, so once it's set, it's retained, even across power downs.

**** START CRT ****

```
A>BAUDSET

Baud rate setting utility for QX-10 Serial Port.
Version 1.10 07/15/82

Current baud rate is: 1200

Enter standard baud rate desired : 4800
Port is now initialized.

A>
```

**** END CRT ****

In order to facilitate the setting of a new baud rate, the BAUDSET utility has been included in TPM-II. To invoke the utility, enter

```
A>BAUDSET
```

The current setting will be displayed, along with a prompt instructing you to enter the new baud rate. The following baud rates will be accepted by the QX-10: 110, 150, 300, 600, 1200, 2400, 4800, and 9600. The example shown above changes the rate to 4800 baud.

END CHAPTER 6