APPENDIX G: TPM BASICS

WHEN TO READ THIS SECTION

The whole point of Valdocs is to create an environment that
provides the *power* of a computer without the need for
traditional "computer literacy". Indeed, many Valdocs users
bought their QX-10 for exactly that reason.

However, *most* Valdocs users would benefit from reading the
sections on *Filenames in TPM, Valdocs Document Filenames* and
Drives and Users. You will run into these things when
"logging into disk drives" and when using copydisk.

Furthermore, the operating system is *there*, all the time,
under the Valdocs Environment, doing its duty. And it *does*
provide quite a bit of power in its own right. This section
provides a very basic introduction to some of that power for
those who are curious about the "computer side" of Valdocs.

WHAT IS TPM-III?

Underneath Valdocs there's a layer of software. known as the
operating system. An operating system manipulates files and
co-ordinates the interactions of the hardware and the
software.

The operating system which Valdocs uses on the QX10 is
called *TPM-III*. TPM stands for *Transient Program Manager*. A
transient program is one which moves in and out of the
system, while the operating system "stays in place".

TPM was written in Z80 "assembly language" for maximum speed
and efficiency. It was first introduced in the mid-1970s,
only a few months after the introduction of CP/M. CP/M was
written in 8080 code, and thus failed to take full advantage
of the then-new Z80 processor. This difference explains why,
in similar system configurations, TPM is somewhat faster
than CP/M.

This speed advantage was one of the reasons TPM was chosen
as the operating system for Valdocs. But in addition, an
advanced "applications environment" like Valdocs requires
features which CP/M does not offer. Having complete control
over TPM allowed us to add features as their need was
discovered.

Thus, TPM-III is one of the most advanced 8 bit operating
systems in use, as well as being among the most mature and
stable as a result of long-term development.

COMPATABILITY BETWEEN TPM III AND CP/M

Most applications designed for use with CP/M Version 2.2 or
Version 1.14 will run under TPM because TPM is *function call
compatible* with CP/M. "Function Calls" are the standard way
a programmer uses the facilities of an operating system.
Therefore, any program which uses CP/M function calls in a
standard way will work just fine under TPM.

However, there is more to it than that. For example, some
programmers use "tricks" and actually have their program
change or move around some of the code in CP/M itself in
order to achieve some effect. Others actually check to *make
sure* that CP/M is really there. And furthermore, versions of
programs which are custom tailored to work with a specific
piece of hardware may or may not run on another piece of
hardware, even when both are running "CP/M". Compatibility
is very complex!

Programmers of major software packages are keenly sensitive
to the problem of "transportability" from machine to
machine, and avoid non-standard programming tricks as much
as possible. Thus, virtually all major software packages, in
their "generic" forms, will work just fine under TPM.

Unfortunatly, the CP/M implementation done for the QX-10
used a wildly different approach to utlizing the special
resources of the QX-10. Thus, CP/M programs customized to
work with that system are not able to run with Valdocs.

To complicate things a bit more, some CP/M programs will run
fine under *raw* TPM, but will not work when Valdocs is
loaded. This occurs because the Valdocs environment is very
sophisticated and creates conditions that extend far beyond
the limitations of the operating systems. The "tricks" may
be acceptable where stand-alone programs are concerned, but
they are deadly in an integrated system.

Programs written for TPM may or may not run under CP/M; it
depends on the intention of the programmer.

If you have difficulty running any CP/M programs under TPM
or in the Valdocs environment, give our Customer Services
Department a call. They can evaluate the nature of the
difficulty and may be able to help you get around it.

To repeat, most *generic* CP/M programs will run just fine
with TPM. Complete Technical Manuals for TPM-III and The
Graphics Driver program are available from Rising Star
Industries.

USER INTERFACES

The user interfaces of TPM and CP/M are very similar, but not identical. If you know how to use CP/M you should be able to find your way around TPM with little difficulty.

However, the *utilities* which are supplied with both packages may operate quite differently, even though the names are similar.

Complete "tool kits" are available from Rising Star, but a few simple utilities are located on the Utilities Disk. Simplified instructions for their use are contained in a file called TPMTOOLS.DOC.

FILE NAMES IN TPM

Valdocs is comprised of a number of *files*. Files can contain either *program information* (instructions for the computer to follow) or *data* (such as letters and pictures).

The name of a file for TPM (or CP/M) is comprised of eight letters or numbers, followed by a period and then three more letters. The first eight letters are called the *filename*. The second three letters are called the *extension*. For example:

        CARDFILE.SYS

Cardfile is the *filename,* SYS is the *extension.*

Extensions are used by programmers to assign files to types. Typical uses of extensions are as follows:

.SYS	A *Program* which belongs to the TPM system
.COM	A *Program* which belongs to the CP/M operating system
.CHN	A *Program* which is part of Valdocs environment
.VAL	Valdocs EDITOR document file
.SPR	Valdocs SPREADSHEET document file
.GRF	Valdocs GRAPH document file
.DAT	Data file
.LOG	Log file
.TMP	Temporary scratch pad file
.PIC	Picture document file
.VCF	Valdocs CARDFILE
.DRW	DRAW document file
.HLP	HELP file
.INI	Initial file (first file used by system after RESET if it is located on Drive A0.)
.OVL	Overlay file used by an application
.MEN	File of menus
.$$$	Temporary file used by TPM when copying files

```
.TXT          Text file (straight text)
.DOC          Document file made by non-Valdocs word
              processors
```

.....and others too numerous to mention. But, as you can see, the letters are usually chosen to tell the person seeing it where the particular file fits in the system.

PRONOUNCING FILENAMES

As an example, the program called BASIC, when displayed as a filename, is written:

BASIC.SYS   (or BASIC.COM)

When spoken out loud, this would be pronounced:

"BASIC dot SYS".

THE FILENAMES OF INDEXED DOCUMENTS

Documents filed in Valdocs have special names assigned to them by the system which indicate the date the file was made (so it's important to set your clock and calendar correctly) and the sequential number of each file made on that date.

A typical filename might be:

84622002.VAL

which can be analyzed as follows:

YYMDD###.XXX   (year, month, day, number. extension).

Our typical filename, 84622002.VAL, thus translates to:

1984, June 22, file #2, EDITOR

Note that there is only one digit for months, but there are 12 months. The tenth month, October, is called A, November is B and December is C.

DRIVES AND USERS

Both floppy disks, hard disks and ramdisks are known as disk drives. These are *physical* drives.

Imagine for a moment that you have a room containing filing cabinets. How do you tell one from another? Usually by naming them. One could be called "A", another "B" and so on.

Physical Disk Drives are like filing cabinets. They are places where data is stored. Since you have more than one of them, they are given names, like: "A", "B", "C"... up to "M".

To continue the analogy, filing cabinets usually have more than one drawer. You can give the drawers any names you want, including numbers. For example, filing cabinet "A" could have drawers numbered 1 through 4. So, if I said that a specific file was in cabinet "A2", you'd know just where to look.

Similarly, disk drives are divided into *user areas*. A user area on a disk drive is analogous to a drawer in a filing cabinet.

The user areas in TPM are numbered from 0 (computer people like to start counting things at zero) to 254.

Thus, if I said that a certain file was on drive A, user 100 (abbreviated to A100), you'd know where to look.

Simple, isn't it?

DRIVE NAMES ON THE QX-10

When only your floppy disks are in use, the LEFT drive is the "A" drive, and the RIGHT drive is the "B" Drive.

In addition, when Valdocs asks for a drive name, you can always refer to your floppy drives by their positions, *LEFT* and *RIGHT*, *no matter what letter name they have.*

A hard disk unit may actually contain one or more disks, called platters, each of which is given a drive name, starting with "A".

The "letter names" of the floppies are whatever letters fall next AFTER the hardisk. I.E., if the hard disk has 4 platters, "A","B","C", and "D", the LEFT floppy would be named "E", and the RIGHT floppy would be named "F".

A Ramdisk is ALWAYS called "M".

HOW MANY FILES IN A CABINET?

On a floppy disk, the maximum number of files on any drive (cabinet) is 128.

On a hard disk, the maximum number of files is either 512 or 1024 (check your hard disk manual).

On a floppy, the absolute minimum size of a file, no matter how little data you actually put in it is 2k, (2000) characters. The minimum size of a file may be different on different physical drives. For example, the minimum file size on the QX-16's ramdisk is 1K.

Files can be placed on any user (in any drawer), but the maximum number of files on one floppy may not exceed 128 *no matter how you divide them up among the user areas.*

The number of permitted files and the minimum size thus interact with each other. That is, although a drive might permit "500K", if it is limited to 128 files, and you put 128 files of 2K each on it, totaling 256K, *that disk is full.*

EXITING VALDOCS TO ENTER TPM

If you declare your level of expertise in SETUP to be advanced or above, a new option appears on MENU. The option is *<E>xit to TPM.*

If you do the above, what you will see is this:

A>█

The "A" tells you what drive you are on. The ">" is called "the prompt". Together they are called the "A prompt". If you were on drive B, you'd see a "B prompt", etc.

The cursor is on the "Command Line". That is, you can type in "commands" to the operating system, instructing it to do things. Whenever you press RETURN, anything you have typed on the command line is passed to the operating system. If what you typed was a valid command, and if it was entered *perfectly*, TPM will then "execute" the command.

If you made even the slightest error in entering the command, it will be refused, and you will be told that you have made a "Syntax Error." (Operating systems are more finicky than *any* high school teacher when it comes to syntax.)

RUNNING PROGRAMS

Probably the most fundamental thing to do on the command line is run a program. Programs have .SYS or .COM extensions.

Just type the NAME of the program, then press RETURN (don't type the extension, just the name).

As you may recall, in MENU, when you select an application
to run, before running the program it puts the program's
name on a line and asks you to supply any needed commands.
When you press return, Menu takes the name of the program
and any commands you typed in and enters them on the TPM
command line for you.

LOGGING DRIVES

Earlier we compared disk drive and user areas to filing
cabinets and drawers. given that simple analogy, If I were
to tell you that a particular file were "on A10" you'd know
where the file is. *But how do you get there?*

When you first exit to TPM, you'll be on a specific drive
and user: usually A0 - which is like "home" to the operating
system. (No matter what drive or user you are on, if you
type the name of a program which is located on A0, and press
return, the program will run, and use whatever drive you are
currently located on for its data.)

To "logging into a drive" is nothing more than *closing one
drawer, and opening another*. How? just type in the letter of
the drive, and number of the user area, followed by a colon
(:), and then press RETURN. For example:

A>B100:(RETURN)

B>█

Closes A0, and logs you into B100.

User numbers are not displayed as part of the B prompt. The
system shows only the B.

If you forget which user area you're on, just press
U(RETURN), and you'll see a message like:

User 100

CHANGING DISKS

To change a floppy disk while in TPM:

1. Insert the new disk in the drive you wish to use.

2. Press CTRL and B.

This is called "logging in the disk". Similar to you logging
"onto" a drive. All of the "Change Data Disk" menus in
Valdocs do this in the background.

It is important to remember, that just as in Valdocs, chang-ing disks without informing the operating system of the fact is courting disaster.

When the "Change Disks" command is given, the system reads the name of every file on the disk, and the *physical* location of each file's data. It then copies all this information into memory where it uses it. If you change disks without "logging them in" the system will not know that the disks are different and will continue to use the information it has in memory. If the system is then asked by some program to write data onto the disk - it will obediently do as told - and proceed to write the data all over the files which already exist on the new disk.

Both the data that was on the disk originally, *and* the data your program tried to write on it are "down the hole". When you hear computer people speak of a "smashed disk", *this* is what they mean.

Always press CTRL B after changing disks!

A FEW SIMPLE COMMANDS

For those who desire complete technical information, the TPM-III manual is available from Rising Star. The intention of this section is to provide the curious Valdocs user with a very brief introduction to some of the more basic TPM commands, which can be put to good use.

DIRECTORY

Format: DIR(CR)

Type DIR followed by a carriage return. What will appear on the screen is a TPM DIRECTORY listing all the filenames on the current drive and user. The number in brackets to the right of each file extension is the *protection level* of the file. A zero means that the file may be erased. A five means that the file cannot be erased.

DIR A100:(CR)

Will show you all the files on Drive A, user 100, from *wherever* you might be..

PROTECTION

Format: PRO FILENAME.EXT <#>(CR)

The protect command allows you to declare a file as undeletable. If you enter a 5, the file cannot be deleted or written to. If you enter a 3, the file can be written to, but only erased by certain programs. If you enter a 0, the file can be either deleted or written to.

If you look at the directory of a Valdocs Data Disk, you will see that some files have different protection levels than others. *Do not change these levels unless you are very very certain as to the results you plan to create.* In particular, *raising* the protection level can cause real havoc - some program expects to be able to write to a file and all of a sudden runs into a computerized stone wall. Crash....

As an example of how the protection command works:

PRO VERSION.SYS <5>(RETURN)

would make the file VERSION.SYS undeletable. You can return it to a form that you can delete by typing:

PRO VERSION.SYS <0>(RETURN)

ERASE FILES

Format: ERA FILENAME.EXT(RETURN)

ERA stands for ERASE, and it means just that. If you type ERA, followed by a filename and a CR, the file will be erased from the disk. Except under *very* limited conditions, once erased, a file is gone forever. SO BE VERY VERY CAREFUL IF YOU ERASE ANYTHING. (If you try to erase a file with protection level 5, TPM will ignore the command and leave the file untouched).

As a word of warning: The utility "DMU" has a "file undelete" command. It actually *can* undelete a file. However, I *strongly* recommend that you not attempt this. The real danger is that a file may appear to have been safely undeleted and then sit like a time bomb waiting to blow your QX-10 back to Japan.

The guiding rule is this:

Rule: If any program has written even one byte of information on the disk since a file was deleted, do not attempt to undelete the file!

The problem with undeleting is this: As soon as a file is deleted, TPM is free to reuse the space that the file occupied. The next time TPM writes anything, that space is fair game. When you undelete a file, it might appear ok, but actually contain parts of other files written since the original was deleted. At that point, you get what are know as "crossed blocks". This means that two files claim to own the same piece of data, which is a big no-no. The only possible way to sort it out is to delete one of the two files claiming ownership of the block - which is rather like asking which thumb you'd like cut off.

And now you know why Valdocs asks "Are you sure?" when you tell it to delete a file.

RENAME FILES

Format: REN NEWFILE.NAM=OLDFILE.NAM(RETURN)

REN means RENAME. Yes, you can actually rename files. Just type REN, followed by the name and extension you want to change the filename to, followed by the old name and extension.

For example:

REN VERSION.OLD=VERSION.SYS(RETURN)

changes the name of the file Version.SYS to Version.OLD.

---------------------- *WARNING* ----------------------------

If you rename any of the files on your Valdocs system disks, sooner or later Valdocs may go looking for the file, fail to find it and...it's hard to say exactly what will happen, but it's not likely to be fun.... So DON'T RENAME Valdocs files unless you know *exactly* why and *what* you're doing.

SPACE AVAILABLE ON DISK

Format: SP(RETURN)

This will give you the space in K (thousands of characters) *and* the number of directory entries unused on the currently logged in Drive.

TYPE A FILE

Format: TY FILENAME.EXT(RETURN)

This will type the contents of a file onto the screen. If you type a CTRL + P, it will toggle your printer on and .DOC, .TXT or .ASC files can be typed to the printer at the same time as they are viewed.

RESTARTING VALDOCS FROM TPM

If you got into TPM via the MENU program, and Valdocs is still loaded and intact, there is an easy way to get *back* to Valdocs:

Format: RESTART(RETURN)

Enter by itself, RESTART will return you to your normal startup module.

If you follow RESTART with the name of one of Valdocs' modules (EDIT, SPRD, DRAW, MAIL, CPYD, etc.) Valdocs will go directly to that module instead of your startup. For example:

RESTART MAIL(RETURN)

would return you to the Valdocs MAIL program.

If the Valdocs environment is non-functional for some reason, or if it hadn't been started in the first place, RESTART will inform you of the fact. In that case, insert your LOAD disk, and press the RESET button.

APPENDIX H: ADVANCED USER TIPS
SPECIAL DISK CONFIGURATIONS

By using the capabilities of COPY DISK, combined with a little knowledge about filenames and the files that comprise Valdocs, it's possible to create "special" Valdocs disks to suit particular needs.

SINGLE-DISK SYSTEMS

A single-disk system is one which will load and run part of Valdocs without having to use 3 separate disks (*Load* and *Run Disks*).

These can be very useful. For example, if you use electronic mail extensively, such a disk would automatically put your system back in MAIL when the power came on after a failure. Similarly, in an office, a single disk EDITOR system might be useful for untrained personnel.

To understand single-disk systems, you must understand the following principle: *In order to run any given module, certain files are mandatory, all others are optional.*

The following are the files that **must** be present in order to LOAD and RUN any Valdocs application:

```
INITIAL.INI
GDRIVER.CHN
HASCI.STL
SYSINIT.CHN
LQ1500.STL or FX80.STL (if one or the other of these
                        printers is in use.)
LOADUP.CHN
INDX.CHN
INDX.OVL
TPM.CHN
```

These are exactly the files which are to be found on your Load Disk. The total space occupied by these files is less than 200K, so on a 376K floppy, you still have about 190K unused in which to place applications of your choice.

By placing a specific application on such a disk and using SETUP to tell the system to START with that application, the special disk will always boot directly into that program without the need for changing disks.

1. Using COPY DISK's *<F>ile Copy, Delete, Rename*
    selection, copy the program(s) you want on your *Single
    Disk System* onto the disk you just prepared.

2. Check the chart in Appendix J to see if the program
    needs any other files in order to work and to spot any
    programs which you feel might be useful on this par-
    ticular disk. If so, copy them now.

    If the program has an overlay or menu file, it MUST
    accompany its main program file. Other associated pro-
    grams are considered desirable.

The following list shows certain program/file combinations
that are mandatory and or useful. More information can be
gleaned from Appendix J.

Program Name	Files used
EDIT.CHN	EDIT.OVL and VALUTIL.CHN
SPRD.CHN	SPRD.OVL
MAIL.CHN	LABELS.SYS
	Address cards are contained in two files: MAILCARD.LOG and MAILINDX.LOG
SCHD.CHN	SCHD.TIM
IUTIL.SYS	IUTIL.MEN, INDX.OVL

3. Enter SETUP and under *Valdocs system characteristics,*
choose the module you want to use as the startup program.

You're now ready to use your single-disk system. Just insert
the single disk in the left drive, a data disk in the right,
and press RESET. The system will start up directly in the
module you've chosen.

REORGANIZING YOUR RUN DISKS

As you saw in the previous section, there are certain
modules that are optional on your system disks. This can be
used to make a single disk system, or it can be used to
re-organize your *Load and Run Disks.*

Example: When you bring up MAIL, the system looks for an address book to use. If it doesn't find one on either the data or the system disk, it asks you where you want to put one. If you have made space on your *Run* (or if you are using single-disk load technique described above), you can select *⟨S⟩ystem disk.* Then, no matter what data disk you use, the address book will be right there ready for use.

Example: Make up a special *Run Disk* that has not only the Editor, but also a dictionary.

Example: If you are using a *RAMDISK* with the system, the overlay files for the Indexer, Editor and Spreadsheet can be automatically loaded into the Ramdisk when loading Valdocs. But *only if they are available on the system disk while Valdocs is loading.* In other words, EDIT.OVL and SPRD.OVL are not on the LOAD disk. Therefore, when using a ramdisk in a floppy based system, these files will only be loaded if you specifically put them on your Load disk.

At the same time, removing these large overlay files from the RUN disk makes lots more room for program modules or Help files on the Run disk.

The following organization of Load and Run Disks has been tried and it works:

Load Disk for use with RAMDISK

INITIAL.INI
LOADUP.CHN
GDRIVER.CHN
HASCI.STL
SYSINIT.CHN
FX80 and/or LQ1500.STL
SETUP.SYS
INDX.CHN
INDX.OVL
EDIT.OVL
SPRD.OVL
TPM.CHN
RESTART.SYS

Run Disk for use with RAMDISK

SETUP.SYS
LABELS.SYS
SCHD.CHN
SCHD.TIM
VERSION.SYS
MENU.CHN
EDIT.CHN
SPRD.CHN

```
MAIL.CHN
DRAW.CHN
CPYD.CHN
TPM.CHN
RESTART.SYS
DEFKEY.SYS
CARDFILE.SYS
```

Note that this organization enables you to get your disk swapping under normal use do to the absolute minimum.

As you can see, by understanding how Valdocs operates, you can organize your Run and Load Disks to suit your particular needs and purposes.

REASSIGNING APPLICATIONS KEYS

When you press an application key (SCHED, CALC, DRAW, EDIT), the system looks for a program module with a specific name, and then runs it. In other words, these keys *point* at a particular program with a certain name.

This too can be put to creative use: Valdocs doesn't care *what* program has what name: these keys just point at a *name*. If you wanted, you could have the DRAW key point at the editor, and the EDIT key point at the graph program!

This can be used to configure the system to fit your personal needs. For example, if you don't use the graphing program (DRAW.CHN) you could rename it to DRAW.SYS so that you could run it from the MENU program. You are then free to name *any* other program DRAW.CHN — and it can then be accessed by that single key.

APPENDIX I:
THE SETUP PROGRAM

The Setup program is used to configure the QX-10 with Valdocs to your exact needs and requirements on a broad variety of topics.

To Access SETUP, press MENU, and select *<S>etup System* off of the main menu. If the SETUP program is not on your system disk, the software will request that you provide a disk which contains it.

When SETUP is first run, the following menu appears:

```
Please select a Setup option and then press RETURN.

    <D>ate and Time                  <C>ommunication Characteristics
    <V>aldocs System Characteristics <T>erminal Characteristics
    <P>rinter Characteristics        <M>iscellaneous peripherals
    <E>xpert options                 <U>ser Identification
                        < UNDO for prior menu >
```

Choices are made from this menu in the normal fashion.

Under each area of choice, the setup options themselves appear as *forms* to be filled out in the document window. The changes entered on such a form do not go into effect until the STORE key is pressed. Thus you are free to experiment with the effects of moving the cursor, making and unmaking choices, etc. (Cursor motion in setup is somewhat different from that in most other modules, and can take some getting used to.)

In many cases you may never need to change the original settings of the options. (One user quipped: "If you don't know what a choice means, you probably don't need to change it.") However, some items, such as printer and modem type might need to be changed often, depending on your operating habits and preferences.

DATE AND TIME

Display Format   Gives a choice of 24 hour
                          (military) or normal display.

Display Clock   Clock *on* or *off* in various
                          modes.

Set New Time   Say YES when you want to change

the displayed time. The cursor
then moves to the line where you
enter the correct time. The time
you enter takes effect at the
moment you press STORE.

Set New Date   Ditto.

VALDOCS SYSTEM CHARACTERISTICS

Experience level   *BEGINNER* means just that.
                            NOVICE implies someone "familiar
                               with the basics of Valdocs
                               as presented in this manual".
                            ADVANCED means "uses Valdocs
                               comfortably and somewhat
                               familiar with TPM file
                               names". (Appendix G)
                            EXPERT means "Knowledgeable about
                               Valdocs and comfortable with
                               TPM and operating systems in
                               general.

At the *Advanced* and *Expert* levels, certain options appear
which are not displayed to less experienced users. Most of
these choices involve the use of TPM filenames, and Advanced
and Expert users are given the ability to exit the Valdocs
environment for the "wilds" of the operating system, where a
user is quite "on his own". Generally, once you are beyond
the basics presented in Section 3 of this manual, feel free
to advance this setting to the highest level. (These choices
were originally incorporated for some features that have yet
to materialize. As it stands, we will probably *rename* the
features by the time they are implemented. So goes
progress.)

Startup Program   When you first enter Valdocs
                            you can instruct the system
                            to come up in whatever program
                            you choose. If you declare
                            yourself to be "expert" in the
                            choice above, the option to
                            start up in TPM will appear.

Indexer Overlay on: If a RAMdisk is in use, do you
                            want the overlay file auto-
                            matically loaded?

Editor Overlay   Ditto.

Spreadsheet Overlay   Ditto.

PRINTER CHARACTERISTICS

These were discussed in detail in Appendix B.

EXPERT OPTIONS

These choices only appear if you declare yourself to be an expert under *System Characteristics*. They offer *plenty* of ways to get yourself in trouble if used without understanding.

Since they all involve *VERY* technical choices that have *nothing* to do with day to day use of Valdocs, an explanation of their meanings is left to the TPM-III MANUAL.

It is *strongly* recommended that you *do not experiment* with these options unless you fully understand their meanings.

COMMUNICATIONS CHARACTERISTICS

Modem Type   Choose the one you have, or none.

Modem Installed?   Even if you have a modem, you
                                may have it turned off. The
                                system needs to know, as certain
                                internal operations depend on
                                the answer.

If no modem is installed, you will see a small "x" to the left of the time display in the lower right corner of the screen.

Duplex Mode   A technical choice involving
                                whether characters sent out over
                                the serial port (RS232) are ex-
                                pected to be *echoed* back by the
                                receiver at the other end of the
                                line or not. Most electronic
                                bulletin boards expect *FULL
                                DUPLEX*, so that is the default.
                                If every character is displayed
                                twice on screen while hooked
                                up person to person, change this
                                to HALF DUPLEX.

Baud Rate	Essentially, *BAUD* rate is the speed at which bits of information are sent over the RS232 line. Both the sending and receiving ends must be the same. The default, 1200, matches the popular *Hayes Smartmodem 1200*.
Data Bits	Another technical choice: 8 and 7 are in common use. Valdocs uses 8.
Stop Bits	Between each character, these act like timing "pauses". In most applications, 1 is the correct number, but it varies.
Parity Mode	Parity is a means of checking the accuracy of data transmission. Used when more elaborate checks are not available. Valdocs Electronic mail, and most popular electronic mail systems use NONE.
Parity Error	When parity is on, and an error is detected, what should the system do? This option displays a special character to show where an error occurred so that it can be corrected.
Background Answer	If you have a modem hooked up to your system, Valdocs has the unique ability to answer the phone and hook up with someone who calls in, even if you're in the middle of doing something else like typing a letter. You can then press the MAIL key to enter mail and receive the communication. If the machine is unattended, it will actually enter Mail *itself* in order to receive the communication. (The screen must be blanked and must *not* be in the INDEXER.) If background MAIL is enabled, an "M" will be displayed at the lower right of the screen next to the time. Background Mail can be toggled on/of with CTRL + STYLE.

Answer text mode	If this is "ON", other mail programs (not just VALDOCS) can call you *Person to Person*, and even if you are not there to answer, can type a text message that will be in your inbasket. If you turn it OFF, only a *file* can be received while the machine is unattended.
Auto Answer Count	How many times do you want the phone to ring before VALDOCS answers? If you use the same phone for voice comm, 3 or 4 might be appropriate. If the line is dedicated, 1 is best. If you set it to ZERO, it won't answer *at all*.

TERMINAL CHARACTERISTICS

Display Length	The screen is capable of 25 lines, some software uses 24. (Valdocs uses 24 for most things)

Display Mode

Exit to TPM mode	Default = Character mode
.COM Selection	Default = Televideo 920
.SYS Selection	Default = Character mode

While all of these choices were available in Valdocs 2, these setup options are far more flexible.

As before, there are three terminal modes available with Valdocs+. Bit mode treats the entire screen as a bit mapped graphic. (This is the *only* mode available on a Color QX-10 by the way. It is used at all times by the Valdocs Editor and Spreadsheet.) Character mode is a normal character display and uses the Valdocs standard. Televideo 920 makes the screen act like this popular terminal.

You can now set the system to shift among these modes in each of three circumstances: when exiting to TPM, when running a .SYS file (a TPM application), and when running a .COM file (a CP/M application).

Thus, Valdocs programs, which are .SYS files, can all run in the correct Valdocs terminal mode. But CP/M programs that have been customized for the QX-10 can be set to run in Televideo mode.

Shift Lock Key	Shift lock works just like a typewriter. ALPHA lock makes only LETTERS uppercase; the row of numbers and punctuation remains unshifted.
Graph-Shift key	This choice allows you to have the Graph-Shift (or ALT) key work in one of two ways: as a shift which effects other keys only while it is held down (the way it usually works), or as an ON/OFF which once pressed, continues to have effect until it is pressed again. Note that Control Key menus are not available when the toggle is ON. See the discussion of Printing Graphics Characters in Appendix B for more information.
Control Key	This toggle allows the CONTROL key to work exactly as above. It was implemented specifically to assist a handicapped user who can only use one finger.
Auto Screen Blank	After 20 minutes unattended, this will cause the screen to blank out. (you can always do it manually by pressing CTRL and STOP.)
Menu Scroll Speed	The interaction windows can either stay fixed, and the text will write inside them, or they can pop up and down at varying rates of speed. Some people like it one way, some the other. We default it to fast.
Scroll Menus	Relating to the above, some people object to the scrolling up and down because it distracts them. But then, the distraction can be a plus when changing between modules because it catches your eye. Again, this is purely a matter of personal taste!

MISCELLANEOUS PERIPHERALS

The system needs to know when you have a plotter and/or a mouse hooked up.

For now, they are only used by the optional *VALDRAW* and *VALPAINT* packages available from Rising Star. In graphics programs the "mouse" is a most enjoyable option.

USER IDENTIFICATION

This is not of any particularly vital importance in the system, and the option disappears after the first time the data is entered. However, the data is accessible by software means and may be used in future software releases. *(We once had a roomful of programmers squabbling over exactly which machine belonged to whom. The USER ID solved it!)*

APPENDIX J:
DISK ORGANIZATION CHART

The following chart lists *all* of the Program modules which are provided as part of the Valdocs system except HELP files. Program modules are found on four disks: LOAD, RUN #1, RUN #2 and UTILITY. (See Section 1 of this manual.)

Some modules are located on more than one disk. Some, such as CPYD and MENU are included because it is convenient for the user to be able to access them from virtually any place in the system. Others, such as INDX.OVL, are REQUIRED on nearly every disk, because *they* are used by *virtually every module*.

Files included on the disk for convenience are marked in parentheses.

When a particular program is required by some other module but only that module, the fact is specifically noted the "Function" section.

PROGRAM NAME	FUNCTION	LOAD	RUN#1	RUN#2	UTILITY
EDIT.CHN	Word Processor		X		
EDIT.OVL			X		
SPRD.CHN	Spreadsheet - used to make electronic calculation sheets			X	
SPRD.OVL				X	
MAIL.CHN	Electronic Mail and Address Book		X		
CARDFILE.SYS	Make user-defined "4x5 File Cards" for any purpose.		X		
LABELS.SYS	Used by MAIL's Addressbook and CARDFILE to "print" data		X		
DRAW.CHN	Business and Scientific Graphs			X	
SCHD.CHN	Appointment Book, Event timer, Itinerary Printouts			X	
SCHD.TIM				X	
DEFKEY.SYS	Used to teach computer to repeat a sequence of keystrokes as a "task"		(x)	(x)	(x)
MENU.CHN	Allows access to Applications that don't have their own keys.		(x)	(x)	(x)
SORT.SYS	SORT (reorder) "ASCII data files" made by CARDFILE or ADDRESSBOOK.		(x)		X
CONVERT.SYS	Convert "ASCII data" from DBASE II, FRIDAY, Peachsofts' List Manager and DIF for use by SORT.				X
VALUTIL.CHN	Convert Version 1 Editor files to Valdocs + format.		(x)		

File	Description				
IUTIL.SYS	Convert Version 1 index files to Version +.				X
IUTIL.MEN					X
MBUILD.SYS	Convert Version 1 Addressbook to Version +.				X
CPYD.CHN	Copy disks, Make data disks, copy from one disk to another		(x)	(x)	(x)
SETUP.SYS	Used to set time, choose printer and modem type, set RS-232 port, select display options, etc.	X		(x)	(x)
VERSION.SYS	Allows user to verify exactly which version of a Valdocs program he is using.				X
RESTART.SYS	Used to re-enter Valdocs when you have exited to TPM operating system.	X	(x)	(x)	X
DMU.SYS	"Disk Management Utility" - expert level utility program				X
ASC.SYS	Allows conversion of Wordstar and Valdocs files to ASCII.				X
LIST.SYS	Utility determines size of files while in TPM				X
ZPIP.SYS	Used to copy files in TPM				X
INITIAL.INI	First program used by TPM operating system when loading Valdocs	X			
LOADUP.CHN	Shows system which files to load in Valdocs system	X			
SYSINIT.CHN	System Support program	X			
GDRIVER.CHN	Graphics and screen display management program	X			
MATRIX.SYS	Used to modify graphic fonts containd in HASCI, LQ1500 and FX-80.STL files				X
HASCI.STL	Contains all fonts used by Valdocs Wordprocessor.	X			(x)
LQ1500.STL	Graphic font used with LQ1500 printer.	X			(x)
FX80.STL	Graphic font used with FX-80 printer.	X			(x)
TPM.CHN	Contains Televideo 920 and Valdocs standard video program.	X	X	X	X
INDX.CHN	Indexed filing system manager	X			
INDX.OVL	Required by indexer, contains print formatter and other vital utilities.	X	X	X	X
QXV361B.SYS	Sysgen program for QX-10				X
Q16V369B.SYS	Sysgen program for QX-16				X

MAKING DISKS FOR THE QX-16

The disks supplied with Valdocs+ are QX-10 System disks.

It is *not ok* to use these disks on the QX-16. You must copy the program modules from the QX-10 disks onto QX-16 disks.

Other than the operating system, the software supplied is identical for the QX-16 and the QX-10. There are some differences in the QX-16HD with a built-in hard disk. Contact Customer Services for details.

1. Boot up in Valdocs 2 on the QX-16.

2. Make a fresh data disk.

3. Use COPYDISK to copy all of the files from
   the QX-10 disks over to the fresh QX16 disks.
   Copy all of the files from the Load Disk and
   both Run disks.

4. From MENU or from TPM, run the program
   entitled "Q16V369B.SYS" off of the Utility
   disk.

Note: If your version of Q16V is higher than 369B (I.E. 369C, 370, etc.) use it. Higher version numbers are more recent.

When running Q16V, it will ask where to put the system. Answer with the letter name of whichever drive holds the floppy with Valdocs+.

5. The new disk should now be a QX-16 System disk.

6. Insert it in your left drive, and press RESET.

7. Copy the disk so that you have both a master and a
   working copy.

QX-16 versus QX-10 disks

The QX-16 can *read* QX-10 disks, however, the QX-10 *cannot* read QX-16 disks. This is due to the simple physical fact that the QX-16 has twice as many "grooves" in which data is written. However, the QX-16 *can* create data disks with the same number of grooves as a QX-10. These can be used to transfer data from the QX-16 to the QX-10, but they should *never* be used for other than data transfer.

Similarly, a data disk made on the QX-10 can be read by the QX-16, but should *never* be used as a data disk. Us it only to *transfer* data to a QX-16.

The reason for all of this is that even when the machines generate the same number of grooves, those grooves are of a different *width,* and if you *write* data to grooves of the wrong width you will get *very unpredictable results.*

Even reformatting a disk is not enough. Once a disk has been made on one machine, to use it on the *other* machine you must erase the disk with a magnet, making it totally blank again.

The rule is: The QX-10 and QX-16 can *read* each others disks. But they should *never* be made to *write* on each other's disks.

APPENDIX K:
HANDLING PROBLEMS

EDITOR FILE GROWTH

The structure of Valdocs+ editor files affords us many
benefits including: the ability to change margins without
having to immediately reform the whole document, the ability
to start working on a document as soon as the cursor
reappears, protection against certain types of loss due to
power outage, the ability to chain out of the editor to
other modules with very small delays, - just to name a few.

However, these benefits are not achieved without their
price. The "virtual document" is organized around a "header"
at the very beginning, which occupies 6K, even in an
otherwise "empty" document. Furthermore, when editing text
in the document, the file grows in chunks. Adding even a
single character, can, under some circumstances, add 500 or
more bytes to the file. Even *deleting* text can increase
file size.

The net result is that it takes neither skill nor talent to
cause a single page document, totaling about 3000 characters
in length, to grow to 5 times that size. To say that some
people find this to be disconcerting is putting it mildly.

Solutions to this have been designed. Where I/O speed
allows it, a tremendous amount of background processing
occurs to minimize the rate of file growth. This can result
in a file shrinking *simply by retrieving it.* Try this.
Retrieve a file. let it stand until ALL disk I/O stops.
Store the file. Retrieve it again. It will have shrunk.

Substantial improvements in these background techniques are
in the works. However, the real value of *these* techniques
will not be widely felt until faster storage devices are
widely available.

However, there are *operating* techniques available to any
user that can result in far better utilization of disk
space.

1. Mark the file as a block and STORE it.

Mark the beginning of your file as a block. Move to the end
of the file, and mark *the entire file as a block*. Then, on
the Control B menu, Store the block as an indexed document.

This results in a file which is its minimum size. For
example, after conversion from ASCII, a file can grow quite

large. Marking the file as a block and storing it can cause
dramatic shrinkage. For example, one of the files in this
manual grew to 62K during editing. When marked as a block,
it shrank to 30K. Similar size reductions are typical.

2. Under Control Q, store the file as a Linear Valdocs File.

In a *Linear Valdocs File,* the text in document is
intersperced with the "imbedded sequences" which control
margins, fonts, underlining and the like. It has *no* 6K
header. This is the minimum size document that will maintain
the Editor's formatting information.

Storing a linear Valdocs file takes about the same amount of
time as it does to store a Block as above - sometimes less.
But retrieving it is always slower, because the Editor must
build the header from scratch. Furthermore, your cursor will
always be at the beginning of the file, and you cannot start
to edit or work on the file until the entire document has
been brought up in the editor.

One technique I use successfully is to retrieve a Valdocs
document, then under control Q, STORE it as a linear file to
its indexed file name. That is, if the indexer filed it
with the name "85823002.VAL", I store it as a linear file to
exactly that same name.

This gives three major benefits:

1. The linear format maximizes space utilization on the
   disk.

2. The linear format is perfect for putting through a spell
   checker such as VSPELL.

3. It can be retrieved back into the editor via the normal
   INDEXER, and one pays only a modest penalty in conversion
   time.

This is so useful that an upcoming release of the Editor
will allow the user to configure the Editor to store this
way whenever the STORE is pressed (*AND* the option will be
available under CONTROL STORE).

ERROR RECOVERY TECHNIQUES

Bugs of all sorts are on the endangered species list for
Valdocs+. Within the next few weeks they will probably be
all but extinct. However, they *can* still occur, either
because you find something we haven't, you have a hardware
problem, or you pull the trip wire on a time-bomb. A comment
on all three situations is appropriate.

1. The Valdocs+ editor contains all new error handling code,
which in the event of an error, invokes its own reformatting
routines which attempt to patch the file.

However, in certain cases a situation could exist that the
reformatter would not be able to repair (yet). In these
cases there are a few error recovery techniques that were
developed early on in the development of Version 2 that
might prove *very* useful.

a. Mark the working part of the document as a block and
    STORE it.

      This is identical to the file shrinkage technique
      mentioned above. If the operation can be done at all,
      the stored block is likely to be in perfect
      condition.

      In V2, we would often find that a document was fine
      down to a certain point, which when touched by the
      system would cause the editor to stop working. The
      recovery technique was to test a couple of times to
      find out just where that spot was, then mark from the
      beginning (or the end) to a point just before (a line
      or two) that spot. If you could get *that* far, you
      usually had it made.

b. Mark the part of the block containing the error as a
    block, and DELETE that.

      This one was less sure, but I once saved a 37 page
      document that wouldn't print by marking a two line
      "something" in the middle as a block and deleting it.
      This is less likely to work, because while marking a
      block, the system must "scan" it, which is the point
      at which most blowups occur in the first place.

c. In cases where the editor hangs up as soon as booted,
    because the restart file (always EDITCURO.VAL in the
    Valdocs Editor) contained something deadly, on many
    occasions I was able to regain control by pressing
    CONTROL W repeatedly while the editor was loading up.
    The reason this sometimes worked was that the editor
    code gives precedence to the *keyboard*. Therefore, if
    the command "goto the top of the file" is the *first*

thing the editor encounters, it may not get to the
mess that tangled it up. *Then* you could attempt
technique A or B above.

d. If you finally give up on a file and conclude that it is
beyond saving, you can get your system out of the
muck by booting up with a fresh data disk, then go
into Copydisk (or TPM if you know how - see Appendix
G), change back to the earlier data disk, and erase
the EDITCURO.VAL file.

While this is drastic, assuming that the blowup is
not one which trashed the entire disk (rare), at
least the remaining files on the disk will again be
usable.

e. Occasionally, if I was *really* desperate, I'd rename the
offending EDITCURO.VAL file to EDITCURO.XXX or some
such. This has the same effect as does erasing it,
but it leaves your file around for study and/or
possible salvage.

f. I cannot overemphasize the importance of good work
habits. No matter *how much* safety and protection we
build into the system there are *still* ways to destroy
it. Your *only* iron clad guarantee of survival is to
make backups.

For example, when I am working on a document longer
than a page or so, I make a backup of the file
everytime I stop for a minute to collect my thoughts.
It's *so* easy: just press STORE, add a "revision X" to
the end of the filename, press STORE again, and press
RETRIEVE twice. Click - click. You can do it without
breaking your train of thought. And if the
unthinkable does occur - you cut your losses.
Understand I am not just talking *bugs:* my two year
old son walked in about 2 months ago and turned off
the power on my hard disk in the middle of receiving
a file via electronic mail. That disk was *trashed.*
Fortunately, I routinely back up my main manuscript
files to *one of my floppies* at least once a work
session - just to be sure.

HARDWARE PROBLEMS

Using an Epson QX-10 can spoil you. You get so used to the
idea that it is unbreakable that you forget that it *can* have
problems. Routine maintainence on your hardware is *really*
important, and by survey, very few of you actually bother to
do it.

It does not take much goop and solidified dust on your drive heads to have them start making errors - and errors have a way of accumulating. First and foremost, *clean your disk drives.* The drive cleaning kits available make it really easy.

Secondly, *clean the fan grill.* About 2 years ago I discovered that the fan grills on a QX-10 will pick up an *amazing* amount of gunk and goop from the air, and after a while, air flow can actually be reduced to ZERO. After my report, people at Epson started checking, and found that it was a general problem. Heat *murders* computers. A hot computer can act *very* strange. (On my machine I remove the grills altogether.)

Third, *get your drives realigned once in a while.* Think of it as a tune-up if you like. QX-10 disk drives use a voice-coil type motor to position the head in and out, and that mechanism is *very* free from wear. You could probably step things in and out a million times and not see any wear - its all done magnetically. *But*, and this is a big but, *electronic components change their values as they age.* Thus, as a electro-mechanical device gets older there are two variables to consider, - wear, *and electrical values.* As these electrical values get further and further away from the nominal factory settings, the chance that they will fall outside the margins increases. There are adjustments that can put this all back in factory spec - but those adjustments can be done only by a trained technician.

This *does* relate to Valdocs: no other piece of software in the world even *begins* to use the facilities of the QX-10 to the same extent as Valdocs. We use every feature, every byte of RAM, and as most of you know, our virtual file system exercises your disks like nothing else. If there *is* a weak link in your hardware, Valdocs will bring it out. Do you really want to find that weak link in the middle of Chapter Two of the "Great American Novel"?

It makes good sense to have your *precision machine* aligned and tuned-up once a year or so, to *maintain* that precision.

3. There is something I call a *time-bomb.* Time-bombs are the result of a blow up that happened long ago - perhaps you didn't even know it - which effects you now in a seemingly random way.

By far the most common type of land mine is an editor document which was damaged invisibly and as a side effect at the same time some *other* editor document blew up. You saw the one blow, it happened right in front of you. You *didn't* see this one blow - because it didn't, till now.

Suspect this when working on older files where a previous blowup occured, and files that seem simple and straight forward just don't want to co-operate. Or where things that usually work *don't*.

For example, say you retrieve a file that *used to be ok* - and all of a sudden it seems to have turned to mush.

Time-bomb....

The problem could be just that one file, or it *could* be a problem with the disk: the media itself could be physically damaged, or its record of data could be somewhat scrambled. In any event, you are better off not taking any chances.

1. Do not use Valdocs on that disk. As we pointed out, Valdocs makes heavy demands on disks, and if a disk error is at fault, Valdocs is *not* the tool with which to fix it.

2. Exit to TPM. (This is why a little TPM knowledge can come in very handy.)

3. Use a program called ZPIP to *copy all of your files over onto a different disk.* This eliminates a whole raft of possible problem areas all in one action.

4. *Then* proceed to attempt the error recovery techniques discussed in the first part of this manual.

The data stored on a disk is organized in a very complex fashion. If *anything* upsets the applecart on the layout of a disk, *all bets are off.* My first response is to get my data (or as much of it as possible) *OFF* of that particular disk, as soon as possible. (I suppose a "backup" could be defined as getting your data off that disk - *before* you needed it.)

BUGS, WISH LISTS, AND DESIGNS

When something in the system just plain doesn't work, we call that a *bug*. So, if you find something that doesn't seem to work, either we didn't explain it well enough, including the limitations, or it really *doesn't* work, in which case it needs fixing.

In either case, we'd like to know about it so that we can fix the bug, or adjust the help files, manuals and menus to better explain things.

In other cases, you'll say, "Gee, I wish I could ...." Well, maybe you *can*, and the manual or help files didn't help you enough, or you can't - in which case we'll consider it for a future release. *Your* idea could show up as a feature next time. (Literally dozens of features in Valdocs+ were the result of user suggestions.)

The third potential problem area is design. We set things up so that they go *BOING*; you'd prefer they go *BONG*. Well, lots of these things boil down to a matter of personal taste, but that doesn't mean that you (or anybody else) is *wrong*. One of the most fundamental principles of *HASCI* (Our computer interface design) is to provide as many interface options as possible so that you can tailor the system to *your* tastes.

Let us know your preferences: *call or write Customer Services.*

CUSTOMER SERVICES DIVISION
RISING STAR INDUSTRIES
25500 HAWTHORNE BLVD. SUITE 2000
TORRANCE, CA. 90505

(213) 373-9127

INDEX

PEACH CALC

DBASE II

PEACHTREE MAIL LIST

FRIDAY

DIF

DATA CONVERT

CHECK BOOK

CARDFILE

SPRD

BUS GRAPH

ASCII CONVERT

ASCII SORT

EDITOR

ADDRESS BOOK

SCHD

VALSPE

EXTERNAL SYSTEMS

MAIL

INDEX

PRINT

CPYD