

## Chapter 9:

### VALDOCS APPLICATIONS

#### MENU

Menu retrieves user-created applications--or any other CPM or TPM programs the user wishes to run--in effect providing an "out" from the Valdocs system (the application is not recognized in some programs). The Menu key always runs the same menu program--it does not give the menu of whatever program the user is in.

The application comes up with runnable files by using TPM instead of Index. Menu is completely user-controlled.

#### CALC

All Valdocs accounting functions are classified under Calc, the system's calculator, which can be accessed by selecting the calculator function in the Editor. Calc is a translation of a Forth BCD floating point package into Stoic. Currently, Calc only provides calculator function.

#### BCD FORMAT

BCD FORMAT: 4 words per BCD number  
13 digits maximum in printing a BCD number  
exponent range is -128 to +127  
4 bits for sign  
4 bits for each of the 13 digits  
8 bits for exponent

#### BCD STORAGE:

Byte 0 : SIGN      DIGIT1  
Byte 1 : EXPONENT  
Byte 2 : DIGIT4    DIGIT5  
Byte 3 : DIGIT2    DIGIT3  
Byte 4 : DIGIT8    DIGIT9  
Byte 5 : DIGIT6    DIGIT7  
Byte 6 : DIGIT12   DIGIT13  
Byte 7 : DIGIT10   DIGIT11

## ASCII STRING TO BCD

The ASCII string that is going to be converted should be put on the stack. SPCT@ is the location of the first digit. Before putting ASCII digits on the stack, set SPCT by typing SPSAVE, which sets SPCT to the current stack pointer. The stack is now ready to receive ASCII digits. No string delimiter or count is needed.

## GFLOAT

After the string is on the stack, type GFLOAT. This executes STRINGTOBCD, which converts the ASCII string on the stack to a BCD number and stores it in BCDTEMP. The stack is not yet cleared off, the DADP is not updated, nor have you yet determined whether or not you have a legal BCD number. These functions are performed by the rest of GFLOAT.

After you execute GFLOAT, the stack will contain five words. TOS = 1 if the number calculated is a proper number, otherwise TOS=0 (TOS indicates Top of Stack). The remaining four words make up the BCD number. An illegal BCD calculation usually results from non-acceptable ASCII characters, or an exponent that is too large.

REMEMBER: THE TOS WORD AFTER EXECUTING GFLOAT TELLS YOU WHETHER OR NOT THE BCD NUMBER IS LEGAL. IF THAT INFORMATION IS UNIMPORTANT, OR YOU KNOW THAT THE NUMBER IS LEGAL, DO "GFLOAT DROP " BEFORE TRYING TO USE YOUR NEW BCD NUMBER.

## DADP

DADP (Digits After Decimal Points) contains the number of digits to be printed after the decimal point. DADP is updated during GFLOAT and is set to the maximum of the number of digits after the decimal point of the number just calculated and the previous value of DADP.

For example: If the first number entered was 8.79, DADP would be 2. If the next number was 6.5557, DADP would be 4. If you then entered 5.6, DADP would still be 4, and the number printed by B= (to print the BCD number) would be 5.6000. DADP has a maximum value of 12.

REMEMBER TO INITIALIZE DADP BEFORE YOU START, OTHERWISE YOU WILL HAVE 12 DIGITS AFTER THE DECIMAL POINT.

## ARITHMETIC

All arithmetic routines require two BCD numbers on the stack and return one BCD number as the result. The first BCD number (TOS-4OS) will be called A; the second BCD number (5OS-8OS) will be called B. The result will be called C.

BCD+ :  $C=A+B$

BCD- :  $C=B-A$

BCD\* :  $C=B*A$

BCD/ :  $C=B/A$

## COMPARISONS

1. With zero: The comparisons expect one BCD number on the stack. They return a 1 if the comparison is true.

BCDO= : TOS=1 if BCD Number = 0 ELSE TOS=0

BCDO< : TOS=1 if BCD Number < 0 ELSE TOS=0

BCDO> : TOS=1 if BCD Number > 0 ELSE TOS=0

2. With other BCD numbers. These comparisons expect two BCD numbers on the stack. Tos-4OS = A, 5OS-8OS =B. They return a 1 if the comparison is true.

BCD= : TOS=1 if  $A=B$ , else TOS=0

BCD< : TOS=1 if  $B<A$ , else TOS=0

BCD> : TOS=1 If  $B>A$ , else TOS=0

## USEFUL WORDS

BCDMIN: Takes two signed BCD numbers and returns the smaller one.

BCDMAX: Takes two signed BCD numbers and returns the larger one.

BCDABS: Takes one BCD number and returns its absolute value.

BCDMINUS: Takes one BCD number and returns the negative.

BCDTOSNGL: Takes one signed BCD number and converts it to a single signed word. BCD numbers larger than 32767

will not convert properly.

**SNGLTOBCD:** Takes a single signed word and converts it to a signed BCD number.

### STACK MANIPULATIONS

**XDUP:** Duplicates BCD numbers already on the stack.  
TOS = How many BCD words to duplicate.

For example: 4 XDUP duplicates the first BCD number.  
Stack starts with 5 words ends with 8 words.

ENTRY: TOS = 4, NOS-5OS = BCD number X  
EXIT: TOS-4OS = one copy of A  
5OS-8OS = second copy of A

8 XDUP duplicated the top two BCD numbers.

ENTRY: TOS=8, NOS-5OS=BCD number A,  
6OS-9OS=BCD number B  
EXIT : TOS-4OS=A, 5OS-8OS=B, 9OS-12OS=A,  
13OS-16OS=B

**4DROP:** Drops four words (or one BCD number) from the stack.

**4SWAP:** Swaps the first two BCD numbers on the stack.

ENTRY: TOS-4OS=A, 5OS-8OS=B  
EXIT: TOS-4OS=B, 5OS-8OS=A

### PRINTING A BCD NUMBER

**BCD@:** Retrieves a BCD number from the address on the stack.

**BCD!:** Stores a BCD number on the stack (NOS-5OS) in the TOS address.

To print a BCD number on the stack, type B=. The B= routine will automatically determine the format for printing the BCD number. If the exponent is out of range for the print routine, the number will be printed in scientific notation. For example, a number with more than 12 digits, such as 12345678998765432, would be printed as 1.234567899876E+16. A very small number will also be printed in scientific notation: .00000000000012345 is printed as 1.2345E-13.

**REMEMBER:**

- 1) Set DADP BEFORE YOU START.
- 2) Set SPCT (do SPSAVE) before you put an ASCII sting on the stack.

## SCHED

### INTRODUCTION

The Schedule calendar of years can be subdivided into months, days, and hours. Each calendar entry is a potential file. SCHED.CHN is written in C, a block-structured programming language similar to Pascal, which utilizes the standard command line. SCHED.CHN has the capability to be called from any module. Currently, however, it can only be called from the Editor because it does not set up temporary save files.

### FILE STRUCTURE

SCHED.CHN utilizes two main files, SCHED.INDX and SCHED.DAT. SCHED.DAT records correspond to days in the appointment book. The header for each record contains a status flag, a Julian date, and an indication of page type.

The page can be an appointment page, a note page 1, or a note page two. The appointment page, which contains the date and a list of hourly appointments, is the default. Pages 1 and 2 jointly form the note page--it is divided into two records for space considerations. The note page is 17 lines by 80 characters: note 1 occupies the top eleven lines, and note 2 the remainder.

The index file points to the records by Julian date. Each Julian date in the index corresponds to the Julian date in the record header. Index size is limited to 400 entries or 100 different days, but when the index is full the records can be saved as one file.

### HEADER DEFINITIONS:

#### SCHD.NDX File Header:

2 bytes=number of index entries (valid entries only--deleted index entries vanish)

#### SCHD.NDX Record or Index Entry:

4 bytes=Julian date

2 bytes=appointment record number pointer into SCHD.DAT

(65000 records total)

2 bytes=record pointer for note 1

2 bytes=record pointer for note 2

(Each record is a day.)

#### SCHD.DAT File Header:

2 bytes=next free record number

2 bytes=next available record number  
1 byte=timer display format  
    0 = displayed as HH:MM:SS  
    1 = displayed as HH.HHH (decimal hours)  
4 bytes=Julian day timer was started (if active)  
4 bytes=seconds into the day that the timer was started  
4 bytes=timer value to display

#### SCHD.DAT Record Header:

1 byte=status flag  
    FFHex=valid in-use record  
    04=deleted record  
    (If reading, it looks for FF--if not found, it  
      comes back with an error message. If adding a record,  
      it looks for 04 to see if it can overwrite that  
      space.)  
4 bytes=Julian date of the record  
1 byte=page type  
    Appointment Page = 0  
    Note 1 = 1  
    Note 2 = 2

The appointment page is the default.

### LOADING AND MEMORY

SCHED.CHN sits on the disk and when called is loaded into bank 0. It first performs a compare for the current date, then displays the day and anything in the appointment page. The data file is loaded into memory and updated one record at a time, so that at most the user will lose the record of that day.

When the user examines a specific day, the appointment page or note page is displayed, and he can do whatever he wishes at that time. 'Store' will save any changes, but the previous data is irretrievable. If the user presses UNDO, nothing will be recovered.

### HELP AND PRINT

SCHED.CHN utilizes the standard help.txt file for its Help functions. The Stop function is not utilized, as there is nothing to stop in SCHED.CHN.

A menu selection of itinerary Print allows the user to select the beginning and ending days to print. The first and last dates are stored temporarily while the Print routine reads and dumps date by date, during which time SCHED.DAT does not use the print spooler. For the purpose of minimal

and speedy Editor functions, an Editor that does not utilize imbedded sequences was created within SCHED.CHN.

#### TIMER

SCHED.CHN provides a timer, contained in the source file SCHED4.C. When initiated, it writes out a small file called SCHEDTIME.DAT and subtracts the last time from the starting time stored there.

#### OTHER FUNCTIONS

Such functions as setting the clock and changing the date are merely system calls. SCHED.CHN provides an option to change the appointment book's time divisions. This option is user dependent, therefore the file, SCHED.TIM, needs no header.

When a bad header is found, the date indexer must be rebuilt through the use of an outside utility called SCHED.REBUILD. Unfortunately, this utility is user-controlled because of lack of space.

#### SCHD ERROR MESSAGES

"index full -- cannot store new page"

Internal date index is full. Make room by deleting old days or by starting a new disk.

"Create/Open/Write" error (text varies)

Indication that the disk or directory is full. Remove files with INDX or start a new disk.

"'get\_free': free record not deleted; rebuild list"

The SCHD.NDX file must be rebuilt with SCHDBILD.

For a full listing of possible error messages, see the TPM manual.

#### DRAW

The DRAW program provides a means to manipulate graphic symbols. Currently, DRAW only provides graphing functions.

DRAW files are retrieved via INDEXER or are created

internally. DRAW "reads" the file, and creates a NAPLPS compatible bit stream, which is then drawn on the screen. DRAW also provides facilities to change the data values and the format of the display.

The DRAW program accomodates all File and Application control keys (e.g., the UNDO key is used to undo out of menus, the PRINT key can be used for a screen print).

DRAW data files are named with a .GRF extension. They consist of ASCII strings separated by a *null* character. Each type of graph has its own format, listed below:

### Bar Graph Format

Offset	Name	Description	Value
0	R.TYPE	TYPE OF GRAPH	"BAR"
1	R.STAMP	VERSION STAMP	"3"
2	R.SHADE	SHADE FOR GRAPH	
3	R.HILO	TITLE POSITION	0 = TOP 1 = BOTTOM
4	R.NAME	NAME OF GRAPH (TITLE)	
5	R.VNAM	VERTICAL AXIS NAME	
6	R.VBEG	VERTICAL AXIS BEGIN VALUE	
7	R.VEND	VERTICAL AXIS END VALUE	
8	R.VDIV	NUMBER OF VERTICAL DIVISIONS	
9	R.HNAM	HORIZONTAL AXIS NAME	
10	R.TORN	HORIZONTAL AXIS TEXT OR NUMBERS FLAG	0 = TEXT LABELS 1 = NUMBERS
11	R.HBEG	HORIZONTAL AXIS BEGIN VALUE \	Only if R.TORN
12	R.HEND	HORIZONTAL AXIS END VALUE /	equals "1"
13	R.HDIV	NUMBER OF HORIZONTAL DIVISIONS (BARS)	
14	R.KIND	"KIND" of Graph	0 = SINGLE
15	R.BAR1	FIRST BAR LABEL FIELD	
...	LABELS	R.HDIV LABELS	
...	DATA	R.HDIV Y DATA VALUES	



## Line Graph Format

Offset	Name	Description	Value
0	R.TYPE	TYPE OF GRAPH	"LINE"
1	R.HILO	TITLE POSITION	0 = TOP 1 = BOTTOM
2	R.NAME	NAME OF GRAPH (TITLE)	
3	R.VNAM	VERTICAL AXIS NAME	
4	R.VBEG	VERTICAL AXIS BEGIN VALUE	
5	R.VEND	VERTICAL AXIS END VALUE	
6	R.VDIV	NUMBER OF VERTICAL DIVISIONS	
7	R.HNAM	HORIZONTAL AXIS NAME	
8	R.TORN	HORIZONTAL AXIS TEXT OR NUMBERS FLAG	0 = TEXT LABELS 1 = NUMBERS
9	R.HBEG	HORIZONTAL AXIS BEGIN VALUE \	Only if R.TORN
10	R.HEND	HORIZONTAL AXIS END VALUE /	equals "1"
11	R.HDIV	NUMBER OF HORIZONTAL DIVISIONS (POINTS)	
12	R.KIND	"KIND" of Graph	0 = SINGLE
13	R.PNT1	FIRST POINT Y VALUE	
...	LABELS	(R.HDIV) LABELS	
...	DATA	(R.HDIV) POINT Y VALUES	

## Scientific Graph Format

Offset	Name	Description	Value
0	R.TYPE	TYPE OF GRAPH	"SCI"
1	R.HILO	TITLE POSITION	0 = TOP 1 = BOTTOM
2	R.NAME	NAME OF GRAPH (TITLE)	
3	R.VNAM	VERTICAL AXIS NAME	
4	R.VBEG	VERTICAL AXIS BEGIN VALUE	
5	R.VEND	VERTICAL AXIS END VALUE	
6	R.VDIV	NUMBER OF VERTICAL DIVISIONS	
7	R.HNAM	HORIZONTAL AXIS NAME	
8	R.TORN	HORIZONTAL AXIS TEXT OR NUMBERS FLAG	0 = TEXT LABELS 1 = NUMBERS
9	R.HBEG	HORIZONTAL AXIS BEGIN VALUE	\ Only if R.TORN
10	R.HEND	HORIZONTAL AXIS END VALUE	/ equals "1"
11	R.HDIV	NUMBER OF HORIZONTAL DIVISIONS (BARS)	
12	R.KIND	"KIND" of Graph	0 = SINGLE
13	R.NPTS	# POINTS IN LINE	
14	R.PNT1	FIRST POINT FIELD	
...	DATA	X \ (R.NPTS) PAIRS Y /	

## Pie Graph Format

Offset	Name	Description	Value
0	R.TYPE	TYPE OF GRAPH	"PIE"
1	R.HILO	TITLE POSITION	0 = TOP 1 = BOTTOM
2	R.NAME	NAME OF GRAPH (TITLE)	
3	R.NUMB	# OF SLICES	
...	PULLOUT FLAG		Y = PULLOUT \ (R.NUMB)
...	SHADE		0-9 = SHADE } TRIPLETS
...	DATA		/

### NOTES

Unfilled sectors are padded with EOF characters (^Z).  
 Version Stamp is either 0, 1, or 3. If 0 or 1, then Version Stamp is actually R.HILO flag. Only bar graphs have a Version Stamp of 3.

## Chapter 10:

### VALDOCS SYSTEM CONTROLS AND UTILITIES

#### SETUP UTILITY

SETUP helps configure the Valdocs system. The configuration is necessary for two reasons: first, to match the hardware to each of the application programs; second, to make the machine perform according to individual preference. "Tailoring" is accomplished via a menu-driven program which changes certain bit patterns in the Clock CMOS RAM.

When the SETUP program is first entered, it reads the "default" settings from the Clock RAM. All changes work on a "temporary" copy until the program is exited via the UNDO key. SETUP checks to see that values were actually changed, writes them in CMOS RAM, and displays the message "NEW VALUES INSTALLED."

The screen and keys are handled differently in SETUP than in other Valdocs modules. SETUP does not allow chaining to other modules via the Applications or File Controls keys. Help for any of the menus can be accessed via the HELP key.

#### HELP

Help is used to clarify confusing situations. All applications use the Help file, called Help.txt, which is accessible only through the SYSINIT module (see SYSINIT under TPM).

#### TO CALL SYSINIT

To call SYSINIT, dec.40 is loaded in the C register, dec. 14 in the B register, the address of the character string in the D E pair register, and dec. 1 in the H register. This is started by calling address 5.

#### HELP MESSAGE FORMAT

The character string must end in a null or binary zero, and begin and end with a colon or semicolon. The format is: carriage return, line-feed, colon, character string, colon. For example:

```
cr lf
:character string:
```

If the help message is more than one page long and you want to allow the user time to read, use a semicolon instead of a colon, e.g.:

;character string;

A colon displays the message: END OF HELP HIT ANY KEY TO RETURN, and the semicolon displays the message: HIT ANY KEY TO CONTINUE. UNDO TO RETURN TO EDITOR.

### SYSINIT OPERATION

Help messages are accessed when the application program delivers a string of characters to SYSINIT. SYSINIT finds the help file and loads it into its own memory, then goes down the file until it finds a character string that matches the one it was given. SYSINIT prints that message onto the screen, until it comes to the next Help message in the file.

In addition to the Help message, SYSINIT will display END OF HELP, HIT ANY KEY TO RETURN. If the Help message is longer than one page, it will pause to allow the user time to read, and display the message HIT ANY KEY TO CONTINUE. UNDO TO EXIT.

### STOP

STOP is the user's panic button. If the user is running several functions simultaneously (such as Print and Mail), STOP will ask which function to terminate.

The Stop program is part of the SYSINIT module, as are Help and Print. When the Stop key is pressed, an 80 Hex is generated, and SYSINIT takes over.

### SYSINIT OPERATION

SYSINIT first determines whether or not printing is active. If the print spooler is inactive, the Stop character (hex code) is passed on to the currently running program. If printing is active, SYSINIT puts a menu on the screen asking the user what he wishes to do.

Because the SYSINIT module takes over as soon as the Stop key is pressed, the aforementioned menu will come up even if an applications program is running. In general, if printing is active, the currently running program will be temporarily stopped. Unfortunately, in MAIL there is a possibility of a "glitch" in the transmitted file where the transmission was temporarily halted. If the stop-menu is left unanswered for too long, the MAIL program will time out.

## COPY DISK

Copy disk allows the user to copy disks, format disks, copy a single file, or use the disk management program, all by doing nothing more than answering the menus.

Copy disk is programmed in the operating system and can be run stand alone in TPM.

## UNDO

Undo is a general function, required in all programs, which allows you to "back out" of a menu. In the Editor, any words, lines, and entire paragraphs can be recovered. Deleted characters or words are stored on a stack. Sentences or larger segments of text are placed in a temporary Undo file.

APPENDIX: A

CMOS CLOCK CHIP RAM LOCATIONS FOR THE EPSON QX-!)  
 ( For VALDOCS Versions 1.14 - 1.18 )

ADDRESS		DEFAULT	FUNCTION	DESCRIPTION
DEC.	HEX	HEX		
0	00	--	SECONDS	0-59 SECONDS
1	01	--	SECONDS/ALARM	--
2	02	--	MINUTES	0-59 MINUTES
3	03	--	MINUTES/ALARM	--
4	04	--	HOURS	0-23 HOURS
5	05	--	HOURS/ALARM	--
6	06	--	DAYS OF WEEK	1-7 DAYS
7	07	--	DAYS OF MONTH	1-31 DAYS
8	08	--	MONTHS	1-12 MONTHS
9	09	--	YEAR	1-99 YEARS
10	0A	2A	REGISTER A	READ/WRITE REGISTER
11	0B	1A	REGISTER B	READ WRITE REGISTER
12	0C	40	REGISTER C	READ ONLY REGISTER
13	0D	80	REGISTER D	READ ONLY REGISTER
14	0E	00	VIDEO SCREEN SET-UP	
			<u>BIT#</u> <u>DISPLAY</u>	
			0        0=12 hr. display, 1=24 hr.	
			1        0=normal clock display, 1=No clock	
( Bits 3-7 undefined )			2        0=normal 25th line in TPM 1=No display (assuming you've selected 24 line)	
15	0F	00	VIDEO EMULATION	
			<u>BIT#</u> <u>EMULATION</u>	
			0 =        R.S.I. drivers	
			1 =        Televideo	
			2 =        Hazeltine	
			3 =        Soroc	
			4 =        I.B.M. PC	
			5 =        Apple-II	
			6 =        I.B.M. 360	
			The lower "nibble" is used to select the desired terminal type.	
			The upper nibble is used as follows:	
			<u>BIT#</u> <u>SELECTION</u>	
			4        0=24 line display, 1=25 line	
			5        0=bit mapped on EXIT to TPM, 1=char mode	
			6        0=bit mapped on .COM file to run, 1=char mode	
			7        0=bit mapped on .SYS file to run, 1=char mode	
16	10	00	SYSTEM DRIVE DEFAULT ( A )	
17	11	00	SYSTEM USER DEFAULT ( 0 )	

ADDRESS		DEFAULT	FUNCTION	DESCRIPTION
DEC.	HEX	HEX		
18	12	01	DATA DRIVE DEFAULT	( B )
19	13	00	DATA USER DEFAULT	( 0 )
20	14	00	SYSTEM DRIVE TEMP	( ALWAYS A )
21	15	00	SYSTEM USER TEMP	
22	16	01	DATA DRIVE TEMP	
23	17	00	DATA USER TEMP	
24	18	00	PRINTER USE	

BIT# SELECTION

0-2 = number of lines to skip for sheet feed  
4set = single sheet bin feed

25 19 00 Editor Flags

BIT# MEANING

0 = set is cursor unlocked  
1 = set is character mode  
2 = set is indexer off

\*Decimal 26-30 and Hex 1A-1E are undefined and reserved for future use.

31 1F C4 MAIL Protocol Options

Bit	Value	Meaning
7-6	--	Character length
	00	5 bits
	01	6 bits
	10	7 bits
	11	8 bits
5	--	Parity error handling
	0	Ignore error, return data as is
4	1	Replace bad char with Off hex
	--	Parity type(affects bit#1 meaning)
3	0	Odd/Even
	1	Mark/Space
	--	Number of stop bits
2	0	1 stop bit
	1	2 stop bits
1	--	Duplex selection
	0	Full duplex mode
0	1	Half duplex mode
	--	Parity value (affected by bit 4)
1	0	Odd (bit 4 = 0)/Space(bit 4 = 1)
	1	Even(bit 4 = 0)/Mark (bit 4 = 1)
0	--	Parity enable
	0	No parity bit sent
	1	Parity bit generated and sent

These protocol options apply only to person-to-person mode. When issuing commands to the Smartmodem, or performing <S>end or <R>eceive functions, the protocol must be set back to the Valdocs standard of 8 data bits, 1 stop bit, no parity.



ADDRESS		DEFAULT	FUNCTION	DESCRIPTION																		
DEC.	HEX	HEX																				
32	20	81	I/O BYTE																			
33	21	33	KEYBOARD REPEAT RATE																			
34	22	00	KEYBOARD STATUS																			
35	23	05	BAUD RATE FOR EXTERNAL SERIAL PORT ( defaults to 1200 baud )																			
		<table border="0"> <tr> <td><u>Top 3 bits define modem</u></td> <td colspan="2"><u>Low nibble defines baud rate</u></td> </tr> <tr> <td>000 = DC Hayes Smart modem</td> <td>0 = 110</td> <td>5 = 1200</td> </tr> <tr> <td>100 = Comrex 300</td> <td>1 = 134.5</td> <td>6 = 2400</td> </tr> <tr> <td>010 = CX 20</td> <td>2 = 150</td> <td>7 = 4800</td> </tr> <tr> <td>011 = other</td> <td>3 = 300</td> <td>8 = 9600</td> </tr> <tr> <td></td> <td>4 = 600</td> <td></td> </tr> </table>			<u>Top 3 bits define modem</u>	<u>Low nibble defines baud rate</u>		000 = DC Hayes Smart modem	0 = 110	5 = 1200	100 = Comrex 300	1 = 134.5	6 = 2400	010 = CX 20	2 = 150	7 = 4800	011 = other	3 = 300	8 = 9600		4 = 600	
<u>Top 3 bits define modem</u>	<u>Low nibble defines baud rate</u>																					
000 = DC Hayes Smart modem	0 = 110	5 = 1200																				
100 = Comrex 300	1 = 134.5	6 = 2400																				
010 = CX 20	2 = 150	7 = 4800																				
011 = other	3 = 300	8 = 9600																				
	4 = 600																					
36	24	01	VIDEO STATUS																			
37	25	06	NULLS ON TTY DEVICE OUTPUT																			
38	26	00	RESERVED FOR SCHEDULER																			
39	27	00	DEBUG AND EXPERT STATUS FLAGS																			
<p><u>Bits 0-1 indicates experience level.</u>  00 = Beginner  01 = Novice  10 = Advanced  11 = Expert</p>																						
<p><u>Debugging Tools</u>  Bit 2 sets Zapple trap. ( via ^ \ command )  Bit 3 sets software debug trap.  ( Sets Graph-shift-STOP Zapple trap.)</p>																						
<p>Bit 4 sets Indexer debug trap.  ( Bits 5-7 are undefined at this time. )</p>																						
40	28	00	MODEM FILE COUNT KEEPER ( Bye )																			
41	29	00	RINGS BEFORE AUTO ANSWER -1 ( Mail & Bye )																			
42	2A	00	USED BY SPOOLER ROUTINE ( Printer speed in characters per second. )																			
43	2B	00	RESERVED FOR EDITOR																			
44	2C	05	MODEM - TEMPORARY BAUD RATE (Bit 7 set indicates Comrex modem initialized.)																			

ADDRESS		DEFAULT	FUNCTION	DESCRIPTION
DEC.	HEX	HEX		
45	2D	00	PRINTER TYPE BYTE	
				0-7 indicates dot-matrix printers. 8-15 indicates non-bit image printers.
				<u>Bottom nibble indicates printer type.</u>
			0000 = MX-80	011x = Yet to be defined
			0001 = MX-100	1000 = Comrex CR1
			0010 = FX-80	1001 = Comrex CR2
			0011 = FX-100	1010-1101 = Undefined
			0100 = RX-80	1110 = Other Printers
			0101 = RX-100	1111 = No printer
				Bit 4 set, indicates a printer which cannot prevent a line-feed from following a carriage return.
				Bit 5 set, indicates a buffered printer.
46	2E	00	RESERVED FOR MAIL	
47	2F	00	RESERVED FOR MAIL	

\*Decimal 48-61 and Hex 30-3D are undefined and reserved for future use.

62	3E	AA	INDICATES VALID SETUP COMPLETED
63	3F	**	SEE IF INITIALIZED ( Default changes with different versions of INITIAL.INI to talk with CMOS clock ram.
			Address out is on port 3D Hex Data in or out is on port 3C Hex

**\*\*NOTE\*\*** Interupts must be disabled before reading or writing to a Clockram Location.

To Read a Clockram Location:

```
MVI A,ADDR ; (address of interest)
DI          ; (disable interupts)
OUT 3D H   ; Hex
IN 3C H    ; Hex
EI
```

To Write a Clockram Location:

```
MVI C,VAL  ; (value to write)
MVI A,ADDR ; (address to write)
DI          ; (disable interupts)
OUT 3D H   ; Hex
MOV A,C
OUT 3D H   ; Hex
EI
```

APPENDIX B

QX-10 VIDEO TERMINAL CONTROL AND ESCAPE KEY CODES

( Video Driver Version 3.35 for Valdocs Version 1.18 )

SEQUENCE	VDRIVER CHAR.MODE	VDRIVER BIT MAPPED	TPM-II CHAR.MODE	FUNCTION
<u>CONTROL CODES</u>				
CTRL "G"	X	X	X	BELL
CTRL "H"	X	X	X	CURSOR (left)
CTRL "J"	X	X	X	CURSOR (down)
CTRL "K"	X	X	X	CURSOR (up)
CTRL "L"	X	X	X	CURSOR (right)
CTRL "M"	X	X	X	CARRIAGE RETURN
CTRL "Z"	X	X	X	CLEAR ALL TO SPACE
CTRL "^^"	X	X	X	HOME ***
CTRL "\"	X	X	X	SPACE ***
CTRL ">"	X	X	X	CARRIAGE RETURN ***
CTRL "-"	X	X	X	SPACE ***

SINGLE CHARACTER ESCAPE SEQUENCES \*\*\*\*

ESC "G"	X	X		ALTERNATE BELL
ESC "T"	X	X	X	CLEAR TO END OF LINE
ESC "Y"	X	X	X	CLEAR TO END OF SCREEN
ESC "j"	X	X	X	REVERSE VIDEO ON
ESC "k"	X	X	X	REVERSE VIDEO OFF
ESC "l"		X		UNDERLINE ON
ESC "m"		X		UNDERLINE OFF
ESC " _"	X	X		SECRET ON
ESC " )"	X		X	START HIGH INTENSITY
ESC " ("	X		X	STOP HIGH INTENSITY
ESC " ^"	X		X	START BLINK
ESC " q"	X		X	STOP BLINK
	X	X		SECRET OFF
ESC "0"				RESERVED
ESC "1"				RESERVED
ESC "2"		X		SELECT "SET" MODE
ESC "3"		X		( overstrike on ) SELECT REPLACE MODE
ESC "4"		X		( normal ) COMPLEMENT MODE
ESC "5"				RESERVED
ESC "6"		X		RESET ON
ESC "7"				RESERVED

QX-10 VIDEO TERMINAL CONTROL AND ESCAPE KEY CODES

( Video Driver Version 3.35 for Valdocs Version 1.18 )

SEQUENCE	VDRIVER CHAR.MODE	VDRIVER BIT MAPPED	TPM-II CHAR.MODE	FUNCTION
----------	----------------------	-----------------------	---------------------	----------

TWO CHARACTER ESCAPE SEQUENCES \*\*\*\*

ESC ". " "0"	X	X	X	CURSOR OFF
*ESC ". " "1"	X	X		BLOCK BLINKING
ESC ". " "2"	X	X		BLOCK STEADY
*ESC ". " "3"	X	X		UNDERLINE BLINKING
ESC ". " "4"	X	X		UNDERLINE STEADY
*ESC ". " "5"	X	X		HALF-SIZE BLINKING
ESC ". " "6"	X	X		HALF-SIZE STEADY
*ESC ". " "7"	X	X		HALF-HIGH BLINKING
ESC ". " "8"	X	X		HALF-HIGH STEADY

THREE CHARACTER ESCAPE SEQUENCES \*\*\*\*

ESC "=" "Y" "X"	X	X	X	LOAD CURSOR POSITION
-----------------	---	---	---	----------------------

\* Note that in BIT-MAPPED mode, the cursor will always be steady in the above sequences.

\*\* All attributes are on a character by character basis.

\*\*\* In BIT-MAPPED mode, These control functions are meant to be used with the CURSOR turned OFF. If used with the CURSOR turned ON it will result in the CURSOR disappearing or leaving UNWANTED CURSOR blocks on the screen.

\*\*\*\* On the QX-10, the Margin Release key is used for Escape functions.

APPENDIX C:

COMPARISON CHART: QX-10 AND QC-10 VIDEO TERMINAL CONTROL AND ESCAPE

CODES

(Video Driver Version 3.15 for Valdocs Version 1.14 - 2.0)  
 (QC-10 is Japanese version of QX-10)

CONTROL KEY SEQUENCES

HEX	QX-10 SEQUENCE	FUNCTION	HEX	QC-10 SEQUENCE	FUNCTION
00		NONE	00		NONE
01		NONE	01		NONE
02		NONE	02		NONE
03		NONE	03		NONE
04		NONE	04		NONE
05		NONE	05	CTRL "e"	ERASE TO END LINE
06		NONE	06		NONE
07	CTRL "g"	BELL	07	CTRL "g"	BELL
08	CTRL "h"	BACK SPACE	08	CTRL "h"	BACK SPACE
09		NONE	09	CTRL "i"	TAB
0A	CTRL "j"	CURSOR DOWN	0A	CTRL "j"	LINE FEED
0B	CTRL "k"	CURSOR UP	0B	CTRL "k"	HOME
0C	CTRL "l"	CURSOR RIGHT	0C	CTRL "l"	CLEAR SCREEN & HOME
0D		NONE	0D	CTRL "m"	CARRIAGE RETURN
0E		NONE	0E		NONE
0F		NONE	0F		NONE
10		NONE	10	CTRL "p"	SCREEN UP
11		NONE	11	CTRL "q"	SCREEN DOWN
12		NONE	12		NONE
13		NONE	13		NONE
14		NONE	14		NONE
15		NONE	15		NONE
16		NONE	16		NONE
17		NONE	17		NONE
18		NONE	18		NONE
19		NONE	19		NONE
1A	CTRL "z"	CLEAR ALL TO SPACE	1A	CTRL "z"	ERASE END SCREEN
1B		NONE	1B	CTRL "<"	ESC
1C	CTRL "\"	SPACE	1C	CTRL "\"	CURSOR (right)
1D	CTRL ">"	CARRIAGE RETURN	1D	CTRL ">"	CURSOR (left)
1E	CTRL "^"	HOME	1E	CTRL "^"	CURSOR (up)
1F	CTRL "-"	SPACE	1F	CTRL "-"	CURSOR (down)

\* All attributes are on a character by character basis.

COMPARISON CHART: QX-10 AND QC-10 VIDEO TERMINAL CONTROL AND ESCAPE CODES

( Video Driver Version 3.15 for Valdocs Version 1.14 - 2.0 )

ESCAPE KEY SEQUENCES

QX-10			QC-10		
HEX	SEQUENCE	FUNCTION	HEX	SEQUENCE	FUNCTION
00		NONE	00		NONE
01		NONE	01		NONE
02		NONE	02		NONE
03		NONE	03		NONE
04		NONE	04		NONE
05		NONE	05		NONE
06		NONE	06		NONE
07	ESC "G"	ALTERNATE BELL	07		NONE
08		NONE	08		NONE
09		NONE	09		NONE
0A		NONE	0A		NONE
0B		NONE	0B		NONE
0C		NONE	0C		NONE
0D		NONE	0D		NONE
0E		NONE	0E		NONE
0F		NONE	0F		NONE
10		NONE	10		NONE
11		NONE	11		NONE
12		NONE	12		NONE
13		NONE	13		NONE
14		NONE	14		NONE
15		NONE	15		NONE
16		NONE	16		NONE
17		NONE	17		NONE
18		NONE	18		NONE
19		NONE	19		NONE
1A		NONE	1A		NONE
1B		NONE	1B		NONE
1C		NONE	1C		NONE
1D		NONE	1D		NONE
1E		NONE	1E		NONE
1F		NONE	1F		NONE
20		NONE	20		NONE
21		NONE	21		NONE
22		NONE	22		NONE
23		NONE	23		NONE
24		NONE	24		NONE
25		NONE	25	ESC "%"	ACCESS CGROM DIRECTLY
26		NONE	26		NONE
27		NONE	27		NONE
28	ESC "("	STOP HIGH INTENSITY	28	ESC "("	PASS THROUGH

29	ESC	)"	START HIGH INTENSITY	29	ESC	)"	PASS THROUGH
2A			NONE	2A	ESC	*"	CLEAR SCREEN
2B			NONE	2B			NONE

ESCAPE SEQUENCES

QX-10				QC-10			
HEX	SEQUENCE	FUNCTION		HEX	SEQUENCE	FUNCTION	
2C		NONE		2C		NONE	
2D		NONE		2D		NONE	
2E	ESC ". "	"0" CURSOR OFF		2E		NONE	
	ESC ". "	"1" CURSOR ON (blinking block)					
	ESC ". "	"2" BLOCK CURSOR ON STEADY					
	ESC ". "	"3" UNDERLINE BLINKING					
	ESC ". "	"4" UNDERLINE STEADY					
	ESC ". "	"5" HALF-SIZE BLINKING					
	ESC ". "	"6" HALF-SIZE STEADY					
	ESC ". "	"7" HALF-HIGH BLINKING					
	ESC ". "	"8" HALF-HIGH STEADY					
2F		NONE		2F		NONE	
30	ESC "0"	RESERVED		30	ESC "0"	REVERSE ON	
31	ESC "1"	RESERVED		31	ESC "1"	REVERSE OFF	
32	ESC "2"	SELECT "SET" MODE ( overstrike on )		32	ESC "2"	CURSOR OFF	
33	ESC "3"	SELECT "REPLACE" MODE ( normal )		33	ESC "3"	CURSOR ON	
34	ESC "4"	COMPLEMENT MODE		34	ESC "4"	UNDERLINE	
35	ESC "5"	RESERVED		35	ESC "5"	UNDERLINE OFF	
36	ESC "6"	RESET ON		36	ESC "6"	HIGHLIGHT	
37	ESC "7"	RESERVED		37	ESC "7"	HIGHLIGHT OFF	
38		NONE		38	ESC "8"	BLINK ON	
39		NONE		39	ESC "9"	NON BLINK	
3A		NONE		3A		NONE	
3B		NONE		3B		NONE	
3C		NONE		3C	ESC "<"	PUSH CURSOR POSITION	
3D	ESC "="	"Y" "X"LOAD CURSOR		3D	ESC "="	SET CURSOR POSITION	
3E		NONE		3E	ESC ">"	POP CURSOR POSITION	
3F		NONE		3F		NONE	
40		NONE		40		NONE	
41		NONE		41		NONE	
42		NONE		42		NONE	
43		NONE		43	ESC "C"	CHANGE CHARACTER SET	
44		NONE		44		NONE	
45		NONE		45		NONE	
46		NONE		46		NONE	
47		NONE		47		NONE	
48		NONE		48		NONE	
49		NONE		49		NONE	
4A		NONE		4A		NONE	
4B		NONE		4B	ESC "K"	SET KANJI SIZE	
4C		NONE		4C	ESC "L"	CHANGE CRT COLOR	
4D		NONE		4D		NONE	
4E		NONE		4E		NONE	

4F NONE

4F NONE

ESCAPE SEQUENCES

QX-10			QC-10		
HEX	SEQUENCE	FUNCTION	HEX	SEQUENCE	FUNCTION
50		NONE	50	ESC "P"	SCREEN DUMP
51		NONE	51		NONE
52		NONE	52		NONE
53		NONE	53		NONE
54	ESC "T"	CLEAR TO END OF LINE	54	ESC "T"	ERASE END OF LINE
55		NONE	55		NONE
56		NONE	56		NONE
57		NONE	57		NONE
58		NONE	58		NONE
59	ESC "Y"	CLEAR TO END OF SCREEN	59	ESC "Y"	ERASE END OF SCREEN
5A		NONE	5A		NONE
5B		NONE	5B		NONE
5C		NONE	5C		NONE
5D		NONE	5D		NONE
5E	ESC "^"	START BLINK	5E		NONE
5F	ESC "_"	SECRET ON	5F		NONE
60		NONE	60		NONE
61		NONE	61		NONE
62		NONE	62		NONE
63		NONE	63		NONE
64		NONE	64		NONE
65		NONE	65		NONE
66		NONE	66		NONE
67		NONE	67		NONE
68		NONE	68		NONE
69		NONE	69		NONE
6A	ESC "j"	REVERSE VIDEO ON	6A		NONE
6B	ESC "k"	REVERSE VIDEO OFF	6B		NONE
6C	ESC "l"	UNDERLINE ON	6C		NONE
6D	ESC "m"	UNDERLINE OFF	6D		NONE
6E		NONE	6E		NONE
6F		NONE	6F		NONE
70		NONE	70		NONE
71	ESC "q"	SECRET OFF STOP BLINK	71		NONE
72		NONE	72		NONE
73		NONE	73		NONE
74		NONE	74		NONE
75		NONE	75		NONE
76		NONE	76		NONE
77		NONE	77		NONE
78		NONE	78		NONE
79		NONE	79		NONE
7A		NONE	7A		NONE
7B		NONE	ESC 7B		SECRET ON
7C		NONE	7C		NONE
7D		NONE	ESC 7D		SECRET OFF



7E NONE  
7F NONE

7E NONE  
7F NONE

ESCAPE SEQUENCES

QX-10		QC-10			
HEX	SEQUENCE	FUNCTION	SEQUENCE	HEX	FUNCTION
80		NONE	ESC	80	1 BYTE CHAR <--> 2 BYTE CHAR
81		NONE	ESC	81	1 BYTE CHAR <--> 2 BYTE CHAR
82		NONE	ESC	82	2 BYTE CHAR <--> 1 BYTE CHAR
83		NONE	ESC	83	JIS CODE <--> SHIFT JIS CODE
84		NONE	ESC	84	SHIFT JIS CODE <--> JIS CODE
85		NONE		85	NONE
86		NONE		86	NONE
87		NONE		87	NONE
88		NONE		88	NONE
89		NONE		89	NONE
8A		NONE		8A	NONE
8B		NONE		8B	NONE
8C		NONE		8C	NONE
8D		NONE		8D	NONE
8E		NONE		8E	NONE
8F		NONE		8F	NONE
90		NONE	ESC	90	PARTIAL SCROLL UP
91		NONE	ESC	91	PARTIAL SCROLL DOWN
92		NONE	ESC	92	SCROLL RIGHT "n" CHAR.
93		NONE	ESC	93	SCROLL LEFT "n" CHAR.
94		NONE	ESC	94	SET SCROLL STEP
95		NONE	ESC	95	SET SCROLL MODE
96		NONE	ESC	96	SCROLL UP "n" LINES
97		NONE	ESC	97	SCROLL DOWN "n" LINES
98		NONE	ESC	98	SET SCROLL SPEED
99		NONE		99	NONE
9A		NONE		9A	NONE
9B		NONE		9B	NONE
9C		NONE		9C	NONE
9D		NONE		9D	NONE
9E		NONE		9E	NONE
9F		NONE		9F	NONE
A0		NONE	ESC	0A0	INS LED ON
A1		NONE	ESC	0A1	INS LED OFF
A2		NONE	ESC	0A2	CAPS LED ON
A3		NONE	ESC	0A3	CAPS LED OFF
A4		NONE	ESC	0A4	NUM LED ON
A5		NONE	ESC	0A5	NUM LED OFF
A6		NONE	ESC	0A6	XXX LED ON
A7		NONE	ESC	0A7	XXX LED OFF
A8		NONE	ESC	0A8	XXX LED ON
A9		NONE	ESC	0A9	XXX LED OFF
AA		NONE		0AA	NONE
AB		NONE		0AB	NONE
AC		NONE		0AC	NONE
AD		NONE		0AD	NONE
AE		NONE		0AE	NONE
AF		NONE		0AF	NONE

ESCAPE SEQUENCES

QX-10

QC-10

HEX	SEQUENCE	FUNCTION	SEQUENCE	HEX	FUNCTION
B0		NONE	ESC	0B0	FUNCTION KEY CHECK MODE ON
B1		NONE	ESC	0B1	FUNCTION KEY CHECK MODE OFF
B2		NONE		0B2	NONE
B3		NONE		0B3	NONE
B4		NONE		0B4	NONE
B5		NONE		0B5	NONE
B6		NONE		0B6	NONE
B7		NONE		0B7	NONE
B8		NONE		0B8	NONE
B9		NONE		0B9	NONE
BA		NONE		0BA	NONE
BB		NONE		0BB	NONE
BC		NONE		0BC	NONE
BD		NONE		0BD	NONE
BE		NONE		0BE	NONE
BF		NONE		0BF	NONE
C0		NONE	ESC	0C0	NORMAL/ZOOM SET & COLUMN SET
C1		NONE		0C1	NONE
C2		NONE	ESC	0C2	HORIZONTAL DOT LINE WRITE
C3		NONE	ESC	0C3	HORIZONTAL DOT LINE ERASE
C4		NONE	ESC	0C4	VERTICAL DOT LINE WRITE
C5		NONE	ESC	0C5	VERTICAL DOT LINE ERASE
C6		NONE	ESC	0C6	DOT LINE WRITE
C7		NONE	ESC	0C7	PSET / PRESET
C8		NONE	ESC	0C8	CHARACTER SIZE WIDE
C9		NONE	ESC	0C9	CHARACTER SIZE NARROW
CA		NONE		0CA	NONE
CB		NONE	ESC	0CB	SET KEI N MODE
CC		NONE	ESC	0CC	CONTROL GUIDE MASK SET/RESET
CD		NONE		0CD	NONE
CE		NONE		0CE	NONE
CF		NONE		0CF	NONE
D0		NONE	ESC	0D0	DISPLAY MODE SET
D1		NONE	ESC	0D1	SELECT DISPLAY SCREEN
D2		NONE	ESC	0D2	DIRECT DISPLAY OF PHYSICAL SCREEN
D3		NONE	ESC	0D3	SELECT FUNCTION KEY DISPLAY
D4		NONE	ESC	0D4	LOCATE TOP OF SCREEN
D5		NONE	ESC	0D5	LOCATE END OF SCREEN
D6		NONE	ESC	0D6	SELECT CURSOR TYPE
D7		NONE	ESC	0D7	FIND CURSOR
D8		NONE		0D8	NONE
D9		NONE		0D9	NONE
DA		NONE		0DA	NONE
DB		NONE		0DB	NONE
DC		NONE		0DC	NONE
DD		NONE		0DD	NONE
DE		NONE		0DE	NONE
DF		NONE		0DF	NONE

ESCAPE SEQUENCES

QX-10		QC-10			
HEX	SEQUENCE	FUNCTION	SEQUENCE	HEX	FUNCTION
E0		NONE	ESC	0E0	SET DOWNLOAD CHARACTER
E1		NONE		0E1	NONE
E2		NONE		0E2	NONE
E3		NONE		0E3	NONE
E4		NONE		0E4	NONE
E5		NONE		0E5	NONE
E6		NONE		0E6	NONE
E7		NONE		0E7	NONE
E8		NONE		0E8	NONE
E9		NONE		0E9	NONE
EA		NONE		0EA	NONE
EB		NONE		0EB	NONE
EC		NONE		0EC	NONE
ED		NONE		0ED	NONE
EE		NONE		0EE	NONE
EF		NONE		0EF	NONE
F0		NONE	ESC	0F0	KEYBOARD REPEAT ON/OFF
F1		NONE	ESC	0F1	KEYBOARD REPEAT START/STOP
F2		NONE	ESC	0F2	SET KEYBOARD REPEAT TIME
F3		NONE	ESC	0F3	SET ARROW KEY CODE
F4		NONE	ESC	0F4	SET SCROLL KEY CODE
F5		NONE	ESC	0F5	SET CONTROL KEY CODE
F6		NONE	ESC	0F6	CLEAR KEY BUFFER
F7		NONE	ESC	0F7	SET KEY SHIFT
F8		NONE		0F8	NONE
F9		NONE		0F9	NONE
FA		NONE		0FA	NONE
FB		NONE		0FB	NONE
FC		NONE		0FC	NONE
FD		NONE		0FD	NONE
FE		NONE		0FE	NONE
FF		NONE		0FF	NONE

\* All attributes are on a character by character basis.

\*\*\*On the QX-10, the [MAR]gin [REL]ease key is used for ESCAPE functions.

APPENDIX D:

KEYBOARD CONVERSION CHART FOR THE EPSON QX-10

HEX	MATRIX	KEY CAP	MATRIX	HEX	KEY CAP
00	61	c/ @	00	--	(undefined)
01	43	c/ a	01	83	UNDO *
02	37	c/ b	02	--	(reserved)
03	35	c/ c	03	84	STORE *
04	45	c/ d	04	85	RETRIEVE *
05	53	c/ e	05	86	PRINT *
06	46	c/ f	06	87	INDEX *
07	47	c/ g	07	8C	MAIL *
08	48	c/ h	08	--	(reserved)
09	58	c/ i	09	88	MENU *
	77	TAB			
0A	49	c/ j	0A	89	CALC. *
	2F	c/ DEC. TAB			
0B	4A	c/ k	0B	8A	SCHED. *
0C	4B	c/ l	0C	8B	DRAW *
0D	39	c/ m	0D	--	(reserved)
	15	ENTER			
	4E	RETURN			
0E	38	c/ n	0E	8E	BOLD *
0F	59	c/ o	0F	8F	ITALIC *
10	5A	c/ p	10	--	(undefined)
11	51	c/ q	11	--	(undefined)
12	54	c/ r	12	--	(undefined)
13	44	c/ s	13	--	(undefined)
14	55	c/ t	14	--	(undefined)
15	57	c/ u	15	0D	ENTER *
16	36	c/ v	16	2E	. (calc.)*
17	52	c/ w	17	30	0 (calc.)*
18	34	c/ x	18	3D	= (calc.)*
19	56	c/ y	19	36	6 (calc.)*
1A	33	c/ z	1A	35	5 (calc.)*
1B	5C	c/ <	1B	34	4 (calc.)*
	74	MARG. REL.			
1C	6C	c/ \	1C	--	(reserved)
1D	5D	c/ >	1D	--	(reserved)
1E	75	c/ ^	1E	91	STYLE *
1F	6A	c/ - (dash key)	1F	90	SIZE *

( \* denotes non-repeating keys )

HEX	MATRIX	KEY CAP	MATRIX	HEX	KEY CAP
20	32	SPACE BAR	20	--	R.SHIFT
21	76	!	21	--	(undefined)
22	4D	"	22	--	(undefined)
23	62	#	23	--	(undefined)
24	63	\$	24	--	(undefined)
25	64	%	25	33	3 (calc.)*
26	66	&	26	32	2 (calc.)*
27	4D	' (accent mk.)	27	31	1 (calc.)*
28	68	(	28	2B	+ (calc.)*
29	69	)	29	39	9 (calc.)*
2A	2D	* (calc.)	2A	38	8 (calc.)*
	67	* (key bd.)			
2B	28	+ (calc.)	2B	37	7 (calc.)*
	2B	+ (key bd.)			
2C	3A	, (comma)	2C	2D	- (calc.)*
2D	2C	- (calc.)	2D	2A	* (calc.)*
	6A	- (dash key)			
2E	16	. (calc.)	2E	BC	(div. key)*
	3B	. (key bd.)			
2F	4F	/	2F	0A	DEC. TAB*
30	17	0 (calc.)	30	--	L.SHIFT
	69	0 (key bd.)			
31	27	1 (calc.)	31	--	(reserved)
	76	1 (key bd.)			
32	26	2 (calc.)	32	20	SPACE BAR
	61	2 (key bd.)			
33	25	3 (calc.)	33	7A	z
	62	3 (key bd.)		5A	Z
				1A	c/ z
				CE	g/ (z key)
				CF	g+s/ (Z key)
34	1B	4 (calc.)	34	78	x
	63	4 (key bd.)		58	X
				18	c/ x
				AB	g/ (x key)
				B5	g+s/ (X key)
35	1A	5 (calc.)	35	63	c
	64	5 (key bd.)		43	C
				03	c/ c
				D0	g/ (c key)
				D4	g+s/ (C key)

HEX	MATRIX	KEY CAP	MATRIX	HEX	KEY CAP
36	19 65	6 (calc.) 6 (key bd.)	36	76 56 16 D1 D5	v V c/ v g/ (c key) g+s/ (C key)
37	2B 66	7 (calc.) 7 (key bd.)	37	62 42 02 D2 C5	b B c/ b g/ (b key) g+s/ (B key)
38	2A 67	8 (calc.) 8 (key bd.)	38	6E 4E 0E D3 D6	n N c/ n g/ (n key) g+s/ (N key)
39	29 68	9 (calc.) 9 (key bd.)	39	6D 4D 0D E7 E5	m M c/ m g/ (m key) g+s/ (M key)
3A	4C	:	3A	2C C8 CA	, (comma) g/ (, key) g+s/ (, key)
3B	4C	;	3B	2E C9 CB	. (key bd.) g/ (. key) g+s/ (. key)
3C	5C	<	3C	9C 9B	PRIOR LN.* s/ PRIOR PG.
3D	1B 6B	= (calc.) = (key bd.)	3D	9D 99	L. CURSOR* s/ L. MARGIN
3E	5D	>	3E	9E 9A	R. CURSOR* s/ R. MARGIN
3F	4F	?	3F	9F 9B	NEXT LINE* s/ NEXT PAGE
40	61	@	40	--	(undefined)
41	43	A	41	93	TAB-REL.*
42	37	B	42	--	SHIFT LOCK
43	35	C	43	61 41 01 CC CD	a A c/ a g/ (a key) g+s/ (A key)

HEX	MATRIX	KEY CAP	MATRIX	HEX	KEY CAP
44	45	D	44	73	s
				53	S
				13	c/ s
				B2	g/ (s key)
				B3	g+s/ (S key)
45	53	E	45	64	d
				44	D
				04	c/ d
				BC	g/ (d key)
				B4	g+s/ ( D key)
46	46	F	46	66	f
				46	F
				06	c/ f
				B7	c/ (f key)
				B6	g+s/ (F key)
47	47	G	47	67	g
				47	G
				07	c/ g
				E3	g/ (g key)
				F3	g+s/ (G key)
48	48	H	48	68	h
				48	H
				08	c/ h
				E1	g/ (h key)
				E0	g+s/ (H key)
49	58	I	49	6A	j
				4A	J
				0A	c/ j
				EB	g/ (j key)
				F8	g+s/ (J key)
4A	49	J	4A	6B	k
				4B	K
				0B	c/ k
				DF	g/ (k key)
				DC	g+s/ (K key)
4B	4A	K	4B	6C	l
				4C	L
				0C	c/ l
				EF	g/ (l key)
				EC	g+s/ (L key)
4C	4B	L	4C	3A	:
				3B	;
				FF	g/ (; key)
				FC	g+s/ (: key)

HEX	MATRIX	KEY CAP	MATRIX	HEX	KEY CAP
4D	39	M	4D	22	"
				27	'
				D9	g/ (' key)
				C1	g+s/ (" key)
4E	38	N	4E	0D	RETURN*
4F	59	O	4F	2F	/
				3F	?
				F7	g/ (/ key)
				F5	g+s/ (? key)
50	5A	P	50	--	R. CONTROL
51	51	Q	51	71	q
				51	Q
				11	c/ q
				A6	g/ (q key)
				B8	g+s/ (Q key)
52	54	R	52	77	w
				57	W
				17	c/ w
				A7	g/ (w key)
				A8	g+s/ (W key)
53	44	S	53	65	e
				45	E
				05	c/ e
				B0	g/ (e key)
				B9	g+s/ (E key)
54	55	T	54	72	r
				52	R
				12	c/ r
				B1	g/ (r key)
				BA	g+s/ (R key)
55	57	U	55	74	t
				54	T
				14	c/ t
				E2	g/ (t key)
				F2	g+s/ (T key)
56	36	V	56	79	y
				59	Y
				19	c/ y
				F1	g/ (y key)
				F0	g+s/ (Y key)



HEX	MATRIX	KEY CAP	MATRIX	HEX	KEY CAP
57	52	W	57	75	u
				55	U
				15	c/ u
				E9	g/ (u key)
				F9	g+s/ (U key)
58	34	X	58	69	i
				49	I
				09	c/ i
				DE	g/ (i key)
				DB	g+s/ (I key)
59	56	Y	59	6F	o
				4F	O
				0F	c/ o
				F6	g/ (o key)
				EB	g+s/ (O key)
5A	33	Z	5A	70	p
				50	P
				10	c/ p
				FE	g/ (p key)
				FB	g+s/ (P key)
5B	5C	[	5B	BE	(1/2 key)
				BF	s/ (1/4 key)
				8D	c/ (1/2 key)
				D7	g/ (1/2 key)
				DB	g+s/ (1/4 key)
5C	6C	\	5C	3C	<
				5B	[
				1B	c/ <
				7B	g/ { (< key)
				C2	g+s/ ([ key)
5D	5D	]	5D	3E	>
				5D	]
				1D	c/ >
				7D	g/ } (> key)
				C3	g+s/ (] key)
5E	75	^	5E	96	INSERT*
5F	6A	_ (underline)	5F	97	WORD*
60	6A	g/ - (dash)	60	--	GRAPHIC SFT.
61	43	a	61	32	2 (key bd.)
				40	@
				00	c/ 2
				A2	g/ (2 key)
				AE	g+s/ (@ key)

HEX	MATRIX	KEY CAP	MATRIX	HEX	KEY CAP
62	37	b	62	33 23 A3 AF	3 (key bd.) # g/ (3 key) g+s/ (@ key)
63	35	c	63	34 24 A4 A9	4 (key bd.) \$ g/ (4 key) g+s/ (\$ key)
64	45	d	64	35 25 A5 AA	5 (key bd.) % g/ (5 key) g+s/ (% key)
65	53	e	65	36 C4 E6 E4	6 (cents key) g/ (6 key) g+s/ (cents key)
66	46	f	66	37 26 DD DA	7 (key bd.) \$ g/ (7 key) g+s/ (& key)
67	47	g	67	38 2A ED EA	8 (key bd.) * g/ (8 key) g+s/ (* key)
68	48	h	68	39 28 FD FA	9 (key bd.) ( g/ (9 key) g+s/ (( key)
69	58	i	69	30 29 F6 F4	0 (key bd.) ) g/ (0 key) g+s/ (0 key)
6A	49	j	6A	2D 5F 1F 60 C6	- (dash) _ (under) c/ - g/ (dash key) g+s/ (under key)
6B	4A	k	6B	3D 2B 7C C7	= (key bd.) + (key bd.) g/ (= key) g+s/ (+ key)

HEX	MATRIX	KEY CAP	MATRIX	HEX	KEY CAP
6C	4B	l	6C	5C	\
				BD	(degree)
				1C	c/ \
				7E	g/ (\ key)
				CO	g+s/ (degree)
6D	39	m	6D	7F	L. X-OUT*
6E	38	n	6E	94	R. X-OUT*
6F	59	o	6F	95	LINE*
70	5A	p	70	--	L. CONTROL
71	51	q	71	82	COPY DISK*
72	54	r	72	81	HELP*
73	44	s	73	80	STOP*
74	55	t	74	1B	MARGIN REL.
75	57	u	75	BB	(low ^ key)
				5E	^
				1E	c/ (low ^ key)
				A0	g/ (low ^ key)
				AC	g+s/ (^ key)
76	36	v	76	31	1 (key bd.)
				21	!
				A1	g/ (! key)
				AD	g+s/ (! key)
77	52	w	77	09	TAB*
78	34	x	78	92	TAB-SET*
79	56	y	79	--	(undefined)
7A	33	z	7A	--	(undefined)
7B	5C	g/ (< key)	7B	--	(undefined)
7C	6B	g/ (= key)	7C	--	(undefined)
7D	5D	g/ (> key)	7D	--	(undefined)
7E	6C	g/ (\ key)	7E	--	(undefined)
7F	6D	L. X-OUT	7F	--	(undefined)
80	73	STOP	80	--	(undefined)
81	72	HELP	81	--	(undefined)
82	71	COPY DISK	82	--	(undefined)
83	01	UNDO	83	--	(undefined)
84	03	STORE	84	--	R. SHIFT (break)
85	04	RETRIEVE	85	--	R. SHIFT (make)
86	05	PRINT	86	--	L. SHIFT (break)
87	06	INDEX	87	--	L. SHIFT (make)
88	09	MENU	88	--	(undefined)
89	0A	CALC.	89	--	(undefined)
8A	0B	SCHED.	8A	--	L. CONTROL (bk.)
8B	0C	DRAW	8B	--	L. CONTROL (mk.)
8C	07	MAIL	8C	--	GRAPHIC (bk.)
8D	5B	c/ (1/2 key)	8D	--	GRAPHIC (mk.)
8E	0E	BOLD	8E	--	R. CONTROL (bk.)
8F	0F	ITALIC	8F	--	L. CONTROL (mk.)

HEX	MATRIX	KEY CAP	MATRIX	HEX	KEY CAP
90	1F	SIZE	MATRIX VALUES 90-FF RETURN NO		
91	1E	STYLE	CODES.....		
92	7B	TAB-SET	ALL KEYS REPEAT USING THE		
93	41	TAB-REL.	THE CONTROL COMMAND.....		
94	6E	R. X-OUT			
95	6F	LINE			
96	5E	INSERT			
97	5F	WORD			
98	3C	s/ PRIOR PAGE			
99	3D	s/ L. MARGIN			
9A	3E	s/ R. MARGIN			
9B	3F	s/ NEXT PAGE			
9C	3C	PRIOR LINE			
9D	3D	L. CURSOR			
9E	3E	R. CURSOR			
9F	3F	NEXT LINE			

HEX	MATRIX	KEY CAP	HEX	MATRIX	KEY CAP
A0	75	g/ (^ key)	B0	53	g/ (e key)
A1	76	g/ (1 key)	B1	54	g/ (r key)
A2	61	g/ (2 key)	B2	44	g/ (s key)
A3	62	g/ (3 key)	B3	44	g+s/ (S key)
A4	63	g/ (4 key)	B4	45	g+s/ (D key)
A5	64	g/ (5 key)	B5	34	g+s/ (X key)
A6	51	g/ (q key)	B6	46	g+s/ (F key)
A7	52	g/ (w key)	B7	46	g/ (f key)
A8	52	g+s/ (W key)	B8	51	g+s/ (Q key)
A9	63	g+s/ (\$ key)	B9	53	g+s/ (E key)
AA	64	g+s/ (% key)	BA	54	g+s/ (R key)
AB	34	g/ (x key)	BB	75	(+/- key)
AC	75	g+s/ (^ key)	BC	2E	(div. key calc.)
AD	76	g+s/ (! key)	BD	6C	(degree key)
AE	61	g+s/ (@ key)	BE	5B	(1/2 key)
AF	62	g+s/ (# key)	BF	5B	(1/4 key)

HEX	MATRIX	KEY CAP	HEX	MATRIX	KEY CAP
C0	6C	g+s/ (\ key)	D0	35	g/ (c key)
C1	4D	g+s/ (" key)	D1	36	g/ (v key)
C2	5C	g+s/ ([ key)	D2	37	g/ (b key)
C3	5D	g+s/ (] key)	D3	38	g/ (n key)
C4	65	(cents key)	D4	35	g+s/ (C key)
C5	37	g+s/ (B key)	D5	36	g+s/ (V key)
C6	6A	g+s/ (_ key)	D6	38	g+s/ (N key)
C7	6B	g+s/ (+ key)	D7	5B	g/ (1/2 key)
C8	3A	g/ (, key)	D8	5B	g+s/ (1/4 key)
C9	3B	g/ (. key)	D9	4D	g/ (' key)
CA	3A	g+s/ (, key)	DA	66	g+s/ (& key)
CB	3B	g+s/ (. key)	DB	58	g+s/ (I key)
CC	43	g/ (a key)	DC	4A	g+s/ (K key)
CD	43	g+s/ (A key)	DD	66	g/ (7 key)
CE	33	g/ (z key)	DE	58	g/ (i key)
CF	33	g+s/ (Z key)	DF	4A	g/ (k key)

HEX	MATRIX	KEY CAP	HEX	MATRIX	KEY CAP
E0	48	g+s/ (H key)	F0	56	g+s/ (Y key)
E1	48	g/ (h key)	F1	56	g/ (y key)
E2	55	g/ (t key)	F2	55	g+s/ (T key)
E3	47	g/ (g key)	F3	47	g+s/ (G key)
E4	65	g+s/ (cents key)	F4	69	g+s/ ( ) key)
E5	39	g+s/ (M key)	F5	4F	g+s/ (? key)
E6	65	g/ (6 key)	F6	69	g/ (0 key)
E7	39	g/ (m key)	F7	4F	g/ (/ key)
E8	49	g/ (j key)	F8	49	g+s/ (J key)
E9	57	g/ (u key)	F9	57	g+s/ (U key)
EA	67	g+s/ (* key)	FA	68	g+s/ (( key)
EB	59	g+s/ (0 key)	FB	5A	g+s/ (P key)
EC	4B	g+s/ (L key)	FC	4C	g+s/ (: key)
ED	67	g/ (8 key)	FD	68	g/ (9 key)
EE	59	g/ (o key)	FE	5A	g/ (p key)
EF	4B	g/ (1 key)	FF	4C	g/ (; key)

## APPENDIX E:

### HASCI EXTENSIONS

---

Along with the recognition of the HASCI keys expressed on the main keyboard, certain control keys have pre-defined functions as well. Any program claiming to adhere to the HASCI interface standard should honor these Control Code Extensions:

#### User Definable Functions

Control 1 thru Control 0      User definable function keys defined via the CONTROL U menu.

#### Cursor Control and Editing Functions

CONTROL W	Beginning of File
CONTROL E	Up-arrow
CONTROL R	Shift Up-arrow (prior page)
CONTROL T	Delete Word Right
CONTROL Y	Delete Line Right
CONTROL A	Cursor Word Left
CONTROL S	Left Arrow
CONTROL D	Right Arrow
CONTROL F	Cursor Word Right
CONTROL G	Delete Right
CONTROL Z	Bottom of Document
CONTROL X	Down Arrow
CONTROL C	Shift Down Arrow (next page)

## Functional Operations

The following Control Codes should be honored in *all* programs expected to work in the Valdocs environment. They control system wide functions.

CONTROL Q	Quick menu. Allows selection of bit-mapped versus character screen, turns the indexer on-off, allows directory and Save and Retrieve direct to operating system.
CONTROL U	User-defined functions. Allows definition of Control 1 thru 0 as as user functions. If such such functions are appropriate to an application, it should recognize this code. See the editor specs for more details.
CONTROL V	Allows switch between viewscreens. See definition document.
CONTROL M	Miscellaneous. This is <i>critical</i> for all applications. At the least, this <i>must</i> recognize CHANGE DATA DISK, closing all temporary files, then instructing the user to change the data disk.

## MAIN FUNCTION KEYS

Some of the top row function keys have predefined meanings that must be honored in all cases.

CONTROL STOP	Blank the view screen.
CONTROL STORE	SAVE any current work and return the user to the current position.
CONTROL PRINT	Screen dump.
CONTROL STYLE	Caps lock.
CONTROL \	Exit to Zapple Monitor.

## UNASSIGNED KEYS

The following keys are unassigned and may be put to any use needed by any application. Their usage in the Valdocs editor is noted, because similar functions exist in many applications, and where possible, similar mnemonics should be used.

CONTROL I	Index, Cross indexes, Tables of Contents.
CONTROL O	On-screen formatting.
CONTROL P	Page layout functions.
CONTROL H	
CONTROL J	
CONTROL K	Correction - access to dictionary.
CONTROL L	Look for menu. Replace. Goto Page #, etc.
CONTROL ;	
CONTROL '	
CONTROL B	Block functions. Move, Copy, Replace, Store, Transfer.
CONTROL N	Notes -Footnotes, references.
CONTROL ,	
CONTROL .	
CONTROL /	
CONTROL <	
CONTROL >	Repaint screen.



APPENDIX F:

RS-232C PIN ASSIGNMENTS

QX-10	PIN	SMARTMODEM*
SYSTEM GROUND	1	SAME
RS-232 (out)	2	RS232 to (in)
RS-232 (in)	3	RS-232 (out)
REQUEST TO SEND	4	(undefined)
CLEAR TO SEND (in)	5	CLEAR TO SEND (out)
DATA SET READY (in)	6	DATA SET READY(out)
SIGNAL GROUND	7	SIGNAL GROUND
CARRIER DETECT (in)	8	CARRIER DETECT (out)
(undefined)	9	(undefined)
(undefined)	10	(undefined)
REVERSE CHANNEL (out) **	11	(undefined)
(undefined)	12	HIGH SPEED (out)
(undefined)	13	(undefined)
(undefined)	14	(undefined)
TRANSMIT CLOCK (in) **	15	(undefined)
(undefined)	16	(undefined)
RECEIVE CLOCK (in) **	17	(undefined)
(undefined)	18	(undefined)
(undefined)	19	(undefined)
DATA TERMINAL READY (out)	20	DATA TRM READY(in)
(undefined)	21	(undefined)
(undefined)	22	RING DETECT (out)
(undefined)	23	(undefined)
TRANSMIT CLOCK (out) **	24	(undefined)
(undefined)	25	(undefined)
(undefined)	26	(undefined)

(modem assignments are based on the HAYES SMARTMODEM. \*)

\* Trademark of HAYES Microcomputer, Inc.

\*\* QX-10 Functions requiring jumper modifications.