© 1991 Tim Kientzle

Table of Contents

I. Purpose 2

II. Quick Introduction 3

III. Configuring V-Term 5

IV. Starting with V-Term 8

V. Summary of Menu Items 13

VI. The Alternate Text Screen 16

VII. The Keyboard 16

VIII. Terminal Emulations 20

IX. Using the Auto-dialer 24

X. Using the Capture Menu 25

XI. Manipulating Disk Files 26

XII. Manipulating Buffers 28

XIII. File Transfers 30

XIV. Input in V-Term 34

XV. Graphics and Hardware Text Displays 35

XVI. Conference Mode 36

XVII. If You Have Problems 37

XVIII. Appendices 39

A. Technical Description of Terminal Emulations 39

B. Technical Description of ASCII and Line Protocols 46

C. Technical Description of XModem and YModem Protocols 48

D. Parameter File Contents 50

I. Purpose

V-Term was originally developed as a project on the Color Computer 1. This earlier program was written in a combination of Forth and assembly language, and used many of the same ideas, but was never made public. Upon the advent of the Color Computer 3, with a larger text screen, and improved memory, the original design ideas were revamped to fill the personal needs of the author. These include an emulation of the VT100 and VT52 terminals, and full support for the available memory of the Color Computer 3. The VT52 and VT100 emulations were tested through use on UNIX mainframes using the EMACS and VI full-screen editors. The support for multiple files in memory was included to remove the need to constantly save capture buffers and XModem downloads to disk. The multitasking structure was designed to allow V-Term to keep up with serial communications while the user examined files in memory or prepared file transfers, or altered communications settings. Suggestions or comments are always welcome, and can be addressed to the author c/o GIMMESOFT.

Features

- Accurate VT100 and VT52 terminal emulation.
- Highly legible 80-column display using high-resolution graphics screens.
- All 128 ASCII characters accessible from the keyboard.
- Easy-to-use menus remove need to memorize command keys.
- Supports the back-panel serial port up to 2400 baud.
- Supports XModem, YModem, and ASCII file transfers directly to disk or memory.
- Onscreen status line displays free memory, capture buffer status, and elapsed time.
- Can print disk files or buffer files, with settable margins, baud rate, and word wrap.
- Parameter files automatically loaded for each system.

New with Version 3.2

- 15-entry autodialer
- 10 programmable macro keys for each system
- Compatible with Hyper I/O and RGBDOS
- Compatible with Disto Super Controller MEB RS-232 ports
- ASCII file transfer

II. Getting on-line with V-Term quick

We'll assume you're familiar with the basic terminology of data communications. If not, refer to the "Introduction to Data Communications."

First, make a backup of your original V-Term disk, and store the original in a safe place. From now on, the only time you should use the original is to make new backups when your backup copies go bad for some reason. This is a practice you should follow with all new software.

Now, to start V-Term, simply put the backup disk in the drive, and type

LOADM"VT

This will load the V-Term loader program (VT.BIN), which will auto-execute and load the main program (V-TERM.PRG). The main program first tries to load a default configuration file called "DEFAULT.VTM", then tries to load an auto-dialer file. Refer to the chapter "Configuring V-Term" for information on setting up these files. The first time you run V-Term, you will get error messages when V-Term is unable to load these files, but you can ignore those for now.

To get on-line, you'll have to set a few basic parameters. V-Term's menu area occupies the bottom three lines of the display. The first of these three lines is the current menu, the second indicates the status of current menu choices, and the third is some basic status information. The third line is also used for error messages, and for prompts for filenames, etc.

To operate the menus, you hold down the Alt key, and tap either the left/right arrow keys to move the bar cursor, and the up/down arrow keys to select a menu item, or simply tap the appropriate letter to select the menu item immediately. (The "appropriate" letter is indicated by the first capitalized letter in the menu item. So Alt-I selects the Input option on the main menu, and Alt-X selects the eXit option to return from a sub-menu.)

The first parameters you'll want to set are the basic modem parameters. Use Alt-P to select the Parameter menu, then Alt-R to select the RS232 menu. (Note that sub-menus are indicated with a plus '+' sign after the name.) From here, select I/ODevice to choose between the back-panel serial port, the Deluxe RS232 pak (or equivalent), or the Direct Connect Modem Pak (which can also be used for modified RS232 paks, or the second port of the PBJ 2sp pack). Then select Baud to set the Baud rate, Word to set the word length, etc. Most of these options are self-explanatory. When these options are properly set, select eXit (Alt-X) to return to the Parameter menu.

The next set of parameters you'll want to set are the Terminal parameters. These parameters determine how information is displayed on the screen, and how the keyboard is interpreted. Select Terminal to get to the Terminal menu. The five palettes used by V-Term can be set by first selecting Palette to choose the appropriate palette (foreground and background for the menu area and the main display, and the screen border color), and then selecting Color to cycle through the available colors. Use Alt with the up and down arrow keys, and hold down keys to take advantage to V-Term's key repeat feature to make this easier. Now set the terminal emulation you prefer with the Emulation option. Besides the Emulation setting, V-Term has a number of minor settings which can be set with the Mode/Status selections. Use Mode to cycle through the possible settings, and Status to change a setting. A few that may be of immediate interest are: Del Key, which changes the Clear and Left Arrow keys to send DEL rather than BS; BSErase, which causes received backspace characters to erase the character they backspace over; ESC key, which sets the Break key to send ESC rather than Ctrl-C; and Arrow Key, which sets the arrow keys to send the sequences appropriate for arrow keys on the selected terminal Emulation. (Note that when Arrow Key mode is On, you must use Clear as the backspace/delete key rather than the left arrow key.)

Once these settings are finished, you should be ready to go on-line. First, you might want to save these settings so V-Term will use them automatically whenever you start up. Select eXit to return to the Parameter menu, then select Save to save the current settings. At the prompt, type "default" (upper/lower case is unimportant) then <ENTER>. This will save the current settings to the file "DEFAULT.VTM", which will be automatically loaded each time you start V-Term.

For more information on configuring the many options in V-Term to your particular tastes, see the chapters that follow.

Have Fun!!

III. Configuring V-Term

V-Term has a myriad of options and settings that can be used to tailor it to your particular preferences. In this chapter, we will outline some of the possible settings, and explain how to set up V-Term to automatically configure itself for each different system you use. In the last section, we briefly explained how to configure V-Term's RS232 parameters, and the basic settings available. Now, we'll explain some of the other settings, give a brief summary of the Terminal Modes available, and explain how to use the Auto-dialer to automatically configure V-Term and dial a system for you (with an auto-dial modem). First, a quick summary of the Modes available on the Terminal menu:

Mode Name	Default	Explanation
Arrow Key	Off	Sets arrow keys to send control sequences mimicking those on the appropriate terminal.
Autowrap	On	Allows cursor to wrap to next line when it reaches the end of the current line.
BSErase	Off	Causes Back Space to erase previous character.
Carr. Ret.	Off	Received CRs are interpreted as CR/LF
Curs. Blink	On	Causes cursor to blink.
Delete Key	Off	Causes Clear and Left Arrow key to send DEL rather than BS.
Dbl Width	Off	Sets entire screen to use double-width characters.
Escape Key	Off	Causes Break key to send ESC rather than Ctrl-C.
Freeze	Off	Causes screen to freeze on Ctrl-S, unfreeze on Ctrl-Q.
Keyclick	Off	Each keypress sounds a click through the monitor speaker.
Key Repeat	On	Most keys repeat if held down.
Line Feed	Off	Received Line Feeds act as New Lines, <Enter> sends CR/LF.
MarginBell	Off	A bell sounds when a key is pressed with the cursor in the 75th column.
Monochrome	On	Turns off color burst on composite output for viewing on monochrome monitors.
Origin	Off	Affects interpretation of VT100 cursor positioning codes.
PF Key	On	Alt-1 thru Alt-4 sends PF1 thru PF4, Off sends macros
View Contl	Off	Causes control codes to be displayed with caret notation for buffer and disk file 'view'.
Xon/Xoff	On	Allows V-Term to automatically use Xon/Xoff to halt host when V-Term's receive buffer fills.

(Note: for most accurate VT100 emulation, set Delete Key mode On, ESC Key mode On, and Arrow Key mode On.)

The Printer options are pretty self-explanatory. You can set the baud rate to the printer, the four margins (left, right, top, and bottom), the number of lines per page, and columns per line, and you can set whether V-Term uses Word wrap to split long lines, and whether it will send CR/LF at the end of each line rather than just CR (this is needed for some printers). These settings affect all printer output in V-Term. You should be aware that each new printer operation (each new file printed) reinitializes the line counter, so for best operation, you should set the paper to the top of a new page each time you want to print something from V-Term.

The Macro menu allows you to set a string to be sent when you press each of Alt-5 through Alt-0. Each one can be defined separately. Simply select Key to choose the key to define, then select Definition to redefine it. Macro key definitions can include control characters by simply typing the control character on the input line. Even a CR can be included by using Ctrl-M. Press <Enter> to store the new definition. V-Term does not automatically send a CR after sending a macro, so if you want the macro key to include <Enter>, you should include a CR (Ctrl-M) at the end of the macro. As an example, you may want to set Alt-5 to send your username with a single key by defining it with "JOE<CR>". Most control characters will display as a small appropriate symbol on the graphics screen. For example, CR will display as a small CR character.

Another setting you may want to set is the number of tracks for each drive on your system. If you are using ADOS3, or another DOS, then you may have each drive set to use 40 or 80 tracks. This can be set on the Disk menu. (Press Alt-X repeatedly to get to the main VTERM menu, then press Alt-D to select the Disk menu.)

Other settings stored in the configuration files include: default capture buffer name, default disk file extension, preferred file transfer protocol, whether file transfers are from disk or memory, default disk drive, and whether V-Term uses a hardware or software text screen display. Typically, you will want to create one configuration file for each different system you use. For example, I have four configuration files: DELPHI.VTM, CIS.VTM, CAMPUS.VTM, and DEFAULT.VTM (which happens to be the same as CAMPUS.VTM, since that's the system I call most often). Once you have set everything for a specific system, use the Save option on the Parameter menu to create a configuration file.

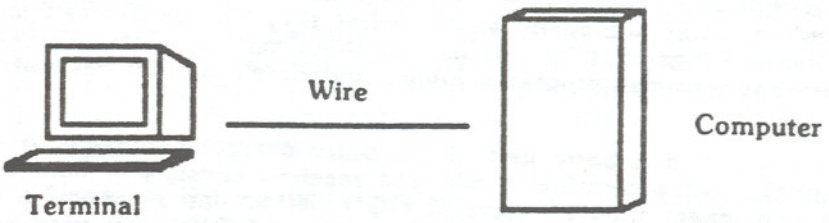
Once you have all of your configuration files set up, you'll want to create an autodial file so that you can easily select the system to call. The Autodial menu has room for 15 systems. Each entry contains a name for the system, a configuration filename, and a string to send to the modem. As an example, select Item until entry number 1 is showing (you can use Alt with the up and down arrows to move both ways through the list), then select Name. You will be prompted for the name of a system. Type a name that's easy for you to remember (like Delphi for Delphi), and press <Enter>. Similarly, use Filename to set the name of a configuration file to associate with that system. Don't include the '.VTM' extension and don't worry about upper/lower case. V-Term will automatically convert the name to all uppercase, and will include the '.VTM' extension when it goes to load the file. Finally, use the ModemString option to define a string to send to the modem to dial this system. For most Hayes-compatible modems, the string will look like 'ATDT555-1234' followed by Ctrl-M (which inserts a CR character). This string can be fairly long, so feel free to include modem set-up commands in the ModemString. Once you have set up each entry that you need, select Save to save the AUTODIAL.VTM file to disk. This file is automatically loaded each time you start V-Term. Now, to dial a system, just select Dial, type the first few characters (no need to type the whole name), and V-Term will search the autodial table, load the configuration file, and dial the modem for you. While it takes a little while to set up V-Term completely, once you do, you'll find that most of the settings (especially the settings on and below the Parameter menu) you should never need to deal with. In fact, if you don't do any uploading or downloading, you might never need to use any menu options except Alt-A Alt-D (to Autodial), and Alt-Q (to quit V-Term).

IV. Introduction to Data Communications

Although the best way to learn about data communications, like any other subject, is through experience over a number of years, there are a number of things that are useful to know as you get started in this. What this section will attempt to do is to explain some of the terms and ideas that are used. We will tell you what a "baud" is for instance, and just what a modem does. So, if you know very little about computer communications, then read on!

What's going on?

Most people encounter computer communications in using a terminal to work on some other computer. In most microcomputers, such as your Color Computer 3, the terminal parts of the computer are built into the rest of the computer. By "terminal parts", we mean the screen and keyboard. That's the idea behind a dedicated terminal, such as the DEC VT100, or the IBM 3270. They are "dumb" terminals that serve simply as a keyboard and screen for another computer, usually a minicomputer or a mainframe. In that case, a diagram of a typical setup looks like this:

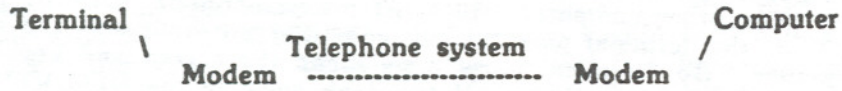


In some cases, the terminal doesn't have a screen, but is rather a "printing" terminal, that looks a lot like a printer with keyboard attached. The terminal's job in this setup is to make it easy for the Computer to talk to a real person. When you press a key on the keyboard, the terminal sends a set of signals over the wire (more on that later) to the computer to tell it which key was pressed, and whenever the computer wants the terminal to display a character, it sends the correct signals over the wire, and the terminal decodes it and prints it on the screen. The idea behind a terminal program such as V-Term is that it enables your Color Computer to understand the signals coming in over the wire (either through the back panel serial port, or the RS232 pak) so that it knows what to display, and to generate the appropriate signals when you press a key on the keyboard. V-Term, of course, is more than just a "dumb" terminal emulator. More on that in a minute, though.

The whys and wherefores of modems

The diagram above is a little unrealistic, because it is rarely possible to put the terminal close enough to the computer to run a wire between them. Usually, we'd like to have the terminal at our home or office so we can use a computer that may be across town, or even on the other side of the world. People quickly realized that an obvious set of wires already exists that connects almost anywhere to almost anywhere else: the telephone lines. There is a slight problem, though. The telephone network is designed to carry the sounds generated by people talking. The signals generated by

computer don't carry well over telephone lines. So modems were invented. The name "modem" is a contraction of the two words "modulator" and "demodulator". A modulator is something that changes a sound according to a certain signal. A demodulator detects the changes in sound to get the original signal back. That's what a modem does: it changes the computer signals into changes in a sound, and detects changes in sound to generate computer signals. A 300 baud modem uses this idea in a very simple way. If you listen to the sound made by a 300 baud modem, it's a steady tone unless it is sending data, in which case the sound changes very rapidly. 1200 baud and 2400 baud modems use the same basic idea, but they don't start with just a simple tone, so you can't tell as much from just listening to them. So now our diagram looks like this:



Bits, Bauds, and other nasty words

Now that you have a basic idea of what a terminal (and a terminal program) is, what a modem is, and what they do, you'll need to know about some things that involve getting them set up right. There are a few settings that terminal programs allow, and here's a brief description of each of them.

Bit - A bit is a basic unit of computer data. It is either on or off. The signals that a terminal sends and receives consists of consecutive bits, with a very exact timing. Several bits together can represent a character. For Baudot code (used in old teletypewriters), 5 bits were used per character. EBCDIC, another older code used only on IBM mainframes, uses 8 bits per character. ASCII, the code designed by the American National Standards Institute (ANSI standard X 3.4), uses 7 bits per character, but is often extended to 8 bits per character. To send one character using the ASCII code requires a start bit (to mark the beginning of the character), either 7 or 8 bits for the character, an optional parity bit (which provides limited error checking), and one or more stop bits (which provide a minimum guaranteed time between characters).

Baud - In typical computer communications, this refers to the number of bits that can be sent in a single second. Thus, 1200 baud modems can send 1200 bits per second, while 300 baud modems can only handle 300 bits per second. The terminal, both modems, and the remote computer must all be using the same speed. Nowadays, most hosts will accept either 300 or 1200 baud, and soon, most will accept 300, 1200, or 2400 baud. To get an idea of how fast these speeds are, some rules of thumb are: Characters per second = baud / 10, Words = characters / 5. So 300 baud is about 30 characters per second, or over 300 words a minute (that's just the possible speed. Most systems don't actually attain that.). No one can type that fast, and few people can read that fast, so 300 baud is fine if you'll only be reading things scrolling up the screen, and typing messages. If you're using full-screen editing, where a lot of data has to be updated very quickly, or doing file transfers, then you'll really appreciate the extra speed of 1200 or 2400 baud.

Word length - This is simply the number of bits used for the actual character data. For the ASCII code, 7 or 8 bits is most common.

Stop bits - These are extra bits added to the end of each character to create some blank time between characters. One or two stop bits are common.

Mark, Space - These are two old telecommunications words, which refer to the two possible line states, on or off. They are also used to refer to the two possible bit values.

Parity bit - Parity checking is a simple form of error detection. An extra bit is calculated and sent along with the character. When received, the calculated value is compared with the received value to check on the accuracy. Parity checking is not necessary when you are using reasonably good-quality phone lines as are used throughout the US, and is not a good enough check to be used for critical communications, so it is slowly fading from general use. Still, many systems do use parity checking. There are two ways to calculate the parity bit, referred to as Even or Odd parity. (Even parity makes the total number of one bits even, Odd parity is similar.) Sometimes the parity bit is permanently set to either Mark or Space. And frequently, the parity bit is not sent at all. It seems common to use a parity bit with 7 bit word lengths, and not to use parity with 8 bit word lengths. The two most common data formats in most areas seem to be 7E1 (7 bit word length, Even parity, one stop bit), or 8N1 (8 bit word length, No parity, one stop bit).

Duplex - Although the original meanings are quite different, this has come to refer to two common ways in which characters you type eventually are displayed on your screen. One way is for the terminal to display all characters. This is referred to as half duplex, and is usually used when two terminals are connected to one another. The other way is for the remote computer to send the characters back to you so they can be displayed. This is referred to as full duplex. If you are using full duplex, and the characters you type don't appear on your screen, then the other computer isn't echoing them back to you, so you should change to half duplex. If you are using half duplex, and the characters are doubled, then the other computer is sending them back, and you should be using full duplex.

RS232pak - One of the difficulties in writing a terminal program is the exact timing necessary in sending and receiving the characters. On the old Color Computers 1 and 2, this has been very difficult, so that 1200 baud could only be achieved by very complicated programming, which would slow down the entire program quite a bit. On the newer Color Computer 3, it is much easier, but it is still difficult to get higher baud rates. The RS232pak contains an Integrated Circuit called an ACIA (Asynchronous Communications Interface Adapter), which does all of the timing automatically. Using an ACIA, it is very easy to keep track of 9600 baud, and still have the program run fast enough to keep track of the screen and other things. We recommend the use of an RS232pak for baud rates above 1200 baud.

Control Characters - Most of the characters in the ASCII code are graphic characters, meaning that they correspond to a certain letter, number, or symbol. The rest of the characters are control characters, which instruct the terminal to perform some action. The most common ones are: LF, line feed, character ten, move down a line; CR, carriage return, character thirteen, return to the beginning of the line; BS, back space, character eight; DEL, delete, character 127, sometimes used like backspace; FF, form feed, character 12, sometimes used as a clear screen code; and ESC, escape, character twenty-seven, which is used to start an escape sequence.

Escape Sequence - The 33 control characters defined by the ASCII standard aren't sufficient to specify all of the actions which terminal makers would like to include in their terminals. So most terminals respond to multi-character control sequences starting with the ESC (escape) control character. Although there have been attempts to standardize these sequences (ANSI standards X 3.41 and X 3.64 especially), most terminal makers invent their own escape sequences for their terminals. (Although the VT100 terminal does follow a part of the X 3.64 standard. And this standard is gaining wider acceptance among microcomputer users since a subset of it is the basis for the "ANSI" standard used in many IBM PC terminal programs.)

Carrier - This refers to the sound generated by the modem when no data is being sent. For 300 baud modems, this is simply a tone. For 1200 baud and 2400 baud modems, it is a "noise". This sound is what the other computer uses to tell that you're modem is on and connected. This is why, if your modem goes off for even a brief time, the other computer will often quickly hang up. This is also the problem with a telephone service called "Call waiting".

Log on, Log off - These refer to the process of starting to talk with the other computer and finishing talking to the other computer. Logging on often consists of typing in your username or usernumber, and some form of password to tell the other computer that you do have the right to use it. Logging off consists of typing 'Bye', or 'Exit', or 'Logoff', or selecting an appropriate menu item.

General Hints and Suggestions

If you've understood the above discussion even vaguely, then you should be prepared to telecommunicate! Just remember a few simple things:

1) Make sure your terminal settings agree with the system you're going to call. If in doubt, try 7 bit word, Even parity, one stop bit, full duplex. Then, if that doesn't work, try 8N1. If you can't see the characters you're typing, try half duplex.

2) There are basically three steps to using any remote system:

   a) Get your terminal program set up.

   b) Dial the number and get your modem set up
      (for auto-dial modems, follow the instructions)

   c) Log on and use the other computer.

3) The biggest difficulty in telecommunicating is learning how to use the other computer. Most BBS's or large Information services have menus to help guide you through. Don't be shy! Experiment! You can't break anything!

V. Summary of Menu Items

V-Term Main Menu

Quit - Terminates the execution of V-Term
Input - Toggles between normal and conference modes
Autodial+ - Autodial sub-menu
Capture+ - Capture sub-menu
Transfer+ - Selects Transfer sub-menu
Buffer+ - Selects Buffer sub-menu
Disk+ - Selects Disk sub-menu
Parameter+ - Selects the Parameter sub-menu

Autodial menu

eXit - Return to V-Term's main menu
Dial - Dial a specific system
Item - Select displayed system
Name - Set name of system entry
Filename - Set name of configuration file for this system
ModemString - Set auto-dial string for modem
Print - Print autodial table to printer
Save - Save autodial table to AUTODIAL.VTM

Capture Menu

eXit - Return to main menu
On/Off - Turn capture buffer On/Off
CopyScreen - Copy current screen into capture buffer
Kill - Kill current capture buffer
Save - Save current capture buffer to disk
View - List current capture buffer to alternate screen
Wait - Pause display of capture buffer
Print - List current capture buffer to printer
BufferName - Set name of buffer to use for capture

Transfer Menu

eXit - Returns to V-Term's main menu
Receive - Receive a file using the selected Protocol
Prompts for the filename and filetype
Transmit - Send a file using the selected Protocol
Prompts for the filename and filetype
Abort - Aborts a protocol transfer in progress
Storage - Select disk or buffer storage
Protocol - Select the Protocol to be used for file transfers

Buffer Menu

eXit - Return to V-Term's main menu

Dir - Sends a directory of the buffers to the alternate text screen

View - Sends a buffer file to the alternate text screen
Selecting View while a file is listing will terminate the list.
Prompts for the Buffer name.

Wait - Pauses a View in progress.
Selecting 'Wait' while a buffer file is listing will pause the display and 'on' will appear beneath 'Wait'. Selecting 'Wait' again will resume the listing.

Print - Sends a buffer file to the printer
Selecting Print while a file is printing will terminate the print.
Prompts for the Buffer name.

Kill - Remove a buffer file from memory
Prompts for the Buffer name, and for verification.

Save - Save a buffer file to a disk file
Prompts for the Buffer name, then the disk file name, then the file type.

Load - Load a buffer file from a disk file
Prompts for the disk file name, then the buffer name.

Mem128k - When 'On', V-Term will only use 128k in a 512k machine. This allows compatibility with various RAMdisks.

Disk Menu

eXit - Returns to V-Term's main menu

Dir - Displays a disk directory to the alternate text screen
Prompts for the disk number, <ENTER> uses the default

View - Displays a disk file to the alternate text screen
Re-selecting View while a view is in progress aborts
Prompts for the file name

Wait - pauses a disk file view in progress

Print - Sends a disk file to the printer
Re-selecting print while a print is in progress aborts
Prompts for the file name

Rename - Renames file on disk
Prompts for the old and new file names

Kill - Removes a disk file from the disk
Prompts for the file name, and verification

SetDrive - Sets the default drive

Tracks - Set the number of tracks per disk.

Extension - Set default file extension for buffer save/load and disk operations.

Cntrlr - Set to Tandy or Disto (If you have the Disto Super Controller. See page 23 Disto Support.)

Parameter Menu

eXit	- Returns to V-Term's main menu
ClockReset	- Resets on-screen timer to 00:00:00
RS232+	- RS232 menu for setting serial port parameters
Terminal+	- Terminal menu for setting screen and keyboard parameters
Macro+	- Macro menu for defining macro key strings
Printer+	- Printer menu for setting printer parameters
Save	- Saves the current parameters to a disk file Prompts for the file name.
Load	- Restores the parameters from a disk file Prompts for the file name.

RS232 Menu

eXit	- Returns to the Parameter menu
Baud	- Cycles the baud rate setting through the available baud rates. Note that the baud rates available depend on the current I/ODevice setting.
Word	- Toggles the RS232 word length between 7 and 8 bits
Stop	- Toggles the number of stop bits between 1 and 2
Parity	- Cycles the parity setting through the available parity settings
Duplex	- Toggles between full and half duplex
I/ODevice	- Toggles between Deluxe RS232pak, Direct Connect Modem pak, Disto (If Cntrlr option on the Disk menu is Disto) and back-panel serial port.

Terminal Menu

eXit	- Returns to the Parameter Menu
Palette	- Selects the Palette to be affected by the 'Color' option
Color	- Cycles the selected Palette through the available colors.
Emulation	- Cycles through the available terminal emulations
Mode	- Selects the screen mode to be affected by the 'State' option
State	- Toggles the selected screen mode on or off. (For more information, see the section on terminal emulations)
Display	- Toggles current screen between the hardware and graphics text displays.
Ver	- Displays the version and copyright notice.

Macro menu

eXit	- Returns to the Parameter Menu
Key	- Select Macro key to define
Definition	- Change the definition of the selected Macro key

Printer Menu

eXit	- Returns to the Parameter Menu
Baud	- Cycles through the available printer baud rates
Columns	- Sets the number of characters per line
Lines	- Sets the number of lines per page
Margin	- Selects the margin to be effected by the 'Size' option
Size	- Sets the corresponding margin
Wrap	- Toggles the Word wrap feature on or off.
UseLF	- When "on", V-Term will follow each CR sent to the printer with a LF, needed for some printers.

Brief notes on operation:

* Any input prompt can be terminated with Break.
* Filename matches are case-insensitive.
* The <F1> key toggles between the two text screens.

VI. The Alternate Text Screen

There are actually two text screens used by V-Term. The first, visible on startup, is the primary communications screen. The other is the alternate screen, used for screen dialogue other than the serial communications. To switch between the two text screens, press <F1>. Most of the menu options which involve text output to the screen (directories, viewing files, and status messages) send their output to the alternate screen. Note that the two screens are completely independent, and that terminal communications are a priority even when the alternate screen is being displayed. So no received characters are lost even when the alternate screen is displayed.

VII. The Keyboard

V-Term interprets the keyboard as consisting of 51 keys plus five "modifiers", keys which change the way other keys are interpreted. The five modifiers are: the two shift keys, the control key, the alt key, and the F2 key. As a general rule, keys pressed with Alt are "local control keys", which cause the program to perform some special action; keys pressed with F2 are "terminal emulation keys", which send special codes associated with keys found on the VT100 terminal which are not found on the standard Color Computer 3 keyboard; and the rest of the keys simply send an appropriate ASCII code when pressed by themselves, or in combination with the Shift or Control keys.

In order to fully understand the way in which V-Term interprets the keyboard, we will look at these three classes of key combinations: local control keys, terminal emulation keys, and others. First, let's look at 'others'. This includes all the keys that are normally used in the terminal communications. First, all of the character keys on the keyboard send the character marked on them as expected, with the shift key working in the way you would expect. Also, the standard control key combinations, (i.e. Ctrl-C) work as expected.

The remainder of the 128 ASCII codes can be sent as follows:

Hex Code	Dec Code	Name	Key combination(s)
0	0	Null	Ctrl-@
8	8	Backspace	Clear ; (see Delete Key mode)
			Ctrl-H ;
			Left arrow (see Arrow Key mode)
9	9	Tab	Ctrl-I
			Right Arrow (see Arrow Key mode)
A	10	Line Feed	Ctrl-J ;
			Down Arrow (see Arrow Key mode)
B	11	Vert. Tab	Ctrl-K ;
			Up Arrow (see Arrow Key mode)
1B	27	Escape	Shft-Break ; Ctrl-Break
1C	28	GS (^\)	Ctrl-<up-arrow>
1D	29	FS (^])	Ctrl-<left-arrow>
1E	30	RS (^^)	Ctrl-<right-arrow>
1F	31	US (^_)	Ctrl-<down-arrow>
5B	91	[	Ctrl-8 [Ctrl-(]
5C	92	\	Ctrl-/
5D	93	]	Ctrl-9 [Ctrl-)]
5E	94	^	Ctrl-7 [Ctrl-']
5F	95	_	Ctrl-<->
60	96	accent grave	Shift-@
7B	123	{	Ctrl-, [Ctrl-<]
7C	124	|	Ctrl-1 [Ctrl-!]
7D	125	}	Ctrl-. [Ctrl->]
7E	126	~	Ctrl-2 ; Ctrl-3 [Ctrl-" ; Ctrl-#]
7F	127	Rubout;Delete (^?)	Ctrl-; (Ctrl-semicolon) ; Ctrl-Clear
			Shift-Clear (see Delete Key mode)

These key combinations have been chosen to be easily remembered. Most follow the conventions used by OS9. The combinations given in brackets are given as a mnemonic aid only. Those in parentheses are common ways of representing certain ASCII control characters (Escape is often represented as Control-[, and in some versions of EMACS, a popular mainframe text editor, the help key is denoted Control-?). NO character requires more than two keys, a modifier (Shift, Alt, Ctrl, or F2) and another key. (Gone are the days of <down-arrow> <shift> <4>!)

Now, for the local control keys. As you already know, <ALT> with the four arrow keys and <ENTER>, and <ALT> with a letter are used to operate the menus. You also already know that <F1> is used to determine which text screen is displayed. Those keys are referred to in this manual as local control keys, since they are used to control the actions of the terminal program. Closely related to these are the Shift-lock keys. To ease usage of V-Term by those people who use OS-9 frequently, V-Term recognizes both Shift-0 and Ctrl-0 as shift-lock toggles. V-Term also recognizes Alt-Clear as a Clear Screen/Terminal Reset key. If the Alternate screen is displayed, Alt-Clear will clear the screen. If the Main Terminal Screen is displayed, then Alt-Clear will clear the screen, and partially reset the terminal, including character set, scrolling region, tab settings, and character attributes.

This is summarized in the following table:

Action	Key combination
Menu cursor left	Alt-(left arrow)
Menu cursor right	Alt-(right arrow)
Select current menu item	Alt-Enter
Increase current menu item	Alt-(up arrow)
Decrease current menu item	Alt-(down arrow)
Select Menu item directly	Alt-(letter)
Shift Lock	Shift-0; Ctrl-0
Switch text screens	F1
Hide/display menus	Alt-F1
Switch Graphics/Hardware	Ctrl-F1
Clear Screen	Alt-Clear
Reset Terminal	Alt-Clear (on Main Terminal Screen only)

Since one of the functions of V-Term is to allow the Color Computer 3 to be used to talk with host computers that can recognize VT100, VT52, or VIDTEX control sequences, it is necessary that V-Term be able to simulate some of the special keys found on these terminals. The diagram below shows the VT100 keypad, which is used extensively by many full-screen editors on large computers, with the corresponding keys for V-Term. These key combinations form a set of keys which send special sequences when pressed, which allow the use of most full-screen editors designed to work with the VT100 or VT52. This set of keys also includes the four arrow keys. The combinations <F2>-<arrow key> will send character sequences which mimic those sent by the arrow keys on the VT100 or VT52 terminals. If the Arrow Key Mode (see below) is on, then the arrow keys without <F2> will also send these sequences. Finally, F2-Break and Alt-Break send true line breaks. F2-Break sends a short line break which functions as an interrupt (like Ctrl-C) on many systems. Alt-Break is a long line break which also deasserts the DTR line to the modem. By doing this, it will hang up most auto-dial modems.

VT100 keypad w/ V-Term equivalents

PF1 * ALT-1	PF2 ALT-2	PF3 ALT-3	PF4 ALT-4
7 F2-7	8 F2-8	9 F2-9	- F2-<->
4 F2-4	5 F2-5	6 F2-6	, F2-<,>
1 F2-1	2 F2-2	3 F2-3	ENTER
0 F2-0		. F2-<.>	F2- ENTER

* The PF1 through PF4 keys are the only exceptions to the general rule that all terminal emulation keys use F2 as a modifier.

Note: For a complete description of the sequences sent by these keys, see Appendix A.

VIII. Description of the Terminal Emulations

What is a Terminal Emulation?

All communications involves a set of basic agreements between the two parties. In normal, day to day life, we take most of this, such as the language used, words that are not used in certain company, etc., very much for granted. When computers communicate, they have to be using the same character set (usually the ASCII character set), and have to agree on the parity, number of bits, baud rate, and so on. For most purposes, this is enough. However, when using your Color Computer to talk with many systems, it's nice if the other system knows how to tell your computer to clear the screen, change to inverse characters, use special characters, etc. Many companies have produced computer terminals which recognize certain codes to perform many of these functions. Two very popular terminals are the VT52 and VT100 terminals produced by the Digital Equipment Corporation, Inc. Because of the success of these terminals, most large systems (including CompuServe and Delphi) know how to use some of the special facilities of these terminals. Many large systems even allow full-screen editing or full-screen games. CompuServe has published its series of Vidtex terminal programs which recognize sequences which are similar to those of the VT52 terminal. V-Term recognizes the most often-used sequences which each of these three terminals use. Some of these sequences direct V-Term to use special characters, to position the cursor on the screen, to erase lines or parts of lines, to scroll all or part of the screen forward or backward, or to do a number of other things (see Appendix A for a full description). Because these sequences are widely known, it is possible to tell many systems that you are using a VT100, VT52, or Vidtex terminal, and they will automatically be able to use many of these special features. Many small bulletin boards assume that all of their users are using Vidtex or VT52 terminals, as these are common and easy to implement.

If you are using a mainframe computer, then having one of these terminal emulations is necessary in order to take advantage of some of the special facilities available on these systems, including full-screen editing, and even windowing on some systems. However, even if you are not using a mainframe computer, having an emulation that your host knows about can allow the host to give you a more pleasing, well-organized display for messages, menus, etc. V-Term also supports a 'CRT' emulation which responds only to a few very common control codes. This corresponds to a 'CRT' or 'Other' terminal on many systems.

Now, although these terminals are well-standardized, many allow the user to change certain aspects of the terminal, and there are some differences in how different Color Computer users prefer their system to work, so V-Term contains several terminal modes, which can be turned on or off to suit the tastes of the individual user. See the section on Parameter files to find out how to have these modes set automatically whenever you start V-Term.

These modes affect control code interpretation, keyboard mapping, and several other aspects of the communication. Some are based upon terminal Modes found in the VT100 terminal, others are unique to V-Term. The default values for these have been chosen to provide the most convenient settings for the majority of users, although individual users should experiment to determine the settings they prefer. The settings which can be set in this fashion are:

Arrow Key Default: Off

When this mode is Off, the four arrow keys are interpreted as the backspace key (left arrow), tab key (right arrow), line feed key (down arrow), and vertical tab key (up arrow). This provides for normal backspace key operation. When this mode is On, the four arrow keys are interpreted as arrow keys, and send special escape sequences depending on the emulation. This is most useful for those using the VT100 or VT52 emulations for full-screen editing. Note that the Clear key is always interpreted by V-Term as a backspace key, regardless of the setting of this mode.

AutoWrap Default: On VT100 Mode

When On, this mode causes characters that would go past the end of a screen line to "wrap" to the next line. When off, they will simply be displayed in the last character position.

BSErase Mode Default: Off

When this mode is On, BS characters erase the positions that are backed over. When Off, BS characters just cause a cursor movement.

Carriage Return Default: Off

When this mode is On, received Carriage Return characters cause a CR/LF to be displayed. This can be useful in half duplex communications.

Cursor Blink Mode Default: On

When on, the cursor blinks approximately once per second.

Delete Key Default: Off

Normally, the Clear key and left arrow key (when the Arrow Key Mode is on) send Back space characters (BS, or character 8) when pressed. Some information services, however, expect a Delete character as a backspace code. When Delete Key Mode is on, the Clear and left arrow keys send Delete characters (Del, or character 127, also called a rubout character). The complete interpretation is indicated in the following chart.

Dbl Width Default: Off

This mode causes V-Term to set the entire Main Terminal screen to double-width characters. This will only display correctly on the Graphics display screen, since double-width characters are not supported on the hardware screen. This mode should not be used with services that do extensive full-screen operations under the VT100 or VT52 emulations, since it does not

allow the full 80-column screen display.

Character sent when Delete Key Mode is:

Key	Off (default)	On
Clear	BS	Del
Ctrl-Clear	Del	BS
left-arrow	BS	Del

Escape Key Default: Off

Normally, the Break key sends Ctrl-C, character 3. When this mode is On, the Break key sends ESC, character 27.

Freeze Default: Off

When this mode is On, V-Term will freeze it's screen display whenever you press Ctrl-S (Xoff) and unfreeze it whenever you press Ctrl-Q (Xon). This can be useful if you are working with either a very fast communications line, or a host system which is slow to respond to Xoff. You should not try to use this with a host which does not recognize Xon/Xoff, since V-Term will then freeze the display, but the host may continue sending and overflow V-Term's receive buffer.

KeyClick Default: Off VT100 Mode

When On, each key pressed will cause a click to sound through the monitor speaker.

Key Repeat Default: On VT100 Mode

When On, most keys will repeat after they have been held down for a short period.

Line Feed Default: Off VT100 Mode

When On, this mode causes received Line Feeds to cause a CR/LF to be displayed, and the Enter key will send a CR/LF pair rather than just a CR. This can often be used when communicating with another computer in half duplex.

MarginBell Default: Off

This mode causes a Bell to sound whenever a key is pressed and the cursor is in column 72. With most services, this will have the effect of sounding the bell when the cursor approaches the margin. However, in full duplex mode, if you type quickly, you may sometimes hear either two or no bells due to a lack of perfect synchronization with the host. This is completely normal.

Monochrome Mode Default: On

When this mode is On, color burst information is removed from the composite video output, resulting in a black and white picture, and a clearer image on most monochrome monitors.

Origin Default: Off VT100 Mode

When On, VT100 cursor positioning codes are interpreted relative to the current scroll window. When Off, they are interpreted relative to the whole screen.

PF Key Default: On VT100 Mode

V-Term now can optionally have 10 programmable macro keys. The Macro menu allows you to program all 10 Alt-<number> combinations. In order to have Alt-1 through Alt-4 send programmed Macro strings rather than the appropriate sequences for the PF1 through PF4 keys on the VT100, you should set PF Key Mode to 'Off'. When On, Alt-1 through Alt-4 are interpreted as PF1 through PF4. When Off, they send the first four macros.

View Control Mode Default: Off

When using the View options to display buffers or disk files, it is sometimes desirable to see exactly what control characters might be in the file. Turning View Control Mode to On will cause all control characters to display with the usual caret notation. (i.e. LFs display as ^J, TABs display as ^I, etc.)

Xon/Xoff Default: On

When this mode is On, V-Term will automatically stop the host whenever its internal receive buffer becomes full. V-Term does this by sending an Xoff character (Ctrl-S) to the host to tell it to stop sending. When the buffer has been depleted, V-Term sends an Xon character (Ctrl-Q) to tell the host to resume. This will work on any system which recognizes Ctrl-Q and Ctrl-S.

Those people who make heavy use of VT100 emulation should probably start by setting Delete Key mode On, Arrow Key mode On, and ESC Key mode On.

Disto Support:

V-Term now supports the Disto MEB RS232 ports. As of this writing, CRC/Disto sells two distinctly different types of RS232 ports. Their RS232 Superpack is a clone of the Tandy RS232 pak, and works well with V-Term set to use the RS232pak. CRC/Disto also markets three add-in boards for use with their Super Controller floppy disk controllers which provide RS232 ports. These ports are mapped very differently from the standard Tandy RS232 pak. To use one of these ports, you must first set the 'Cntrlr' option on the Disk menu to 'Disto', and you then have access to a 'Disto' option under I/ODevice on the RS232 menu.

>>> WARNING <<<

Note: If you do not have a Disto Super Controller, do NOT set the 'Cntrlr' option to 'Disto'. Attempting to select the 'Disto' I/ODevice option with a non-Super Controller will cause erratic disk operation and possibly crash V-Term. V-Term also requires the Disto RS232 add-in to be in the same slot as the Super Controller, since V-Term does not do any slot switching.

IX. Using the Autodialer

V-Term has the ability, through it's parameter files, to be completely configured for each different system you use. However, using the Parameter Save and Load options to manually load a different parameter file for each system you use can be somewhat cumbersome. If you only use V-Term to call one system, then you can set up the 'default' parameter file, which will be automatically loaded each time you start V-Term. But this is obviously not enough if you regularly call up more than one system.

Enter V-Term's Autodialer. From the Autodial menu, you can set up a list of up to fifteen different systems. Associated with each system is the name of the system, the name of a parameter file containing the parameter setup for that system, and a modem string to be sent to the modem. We'll first explain how the autodialer works, and then detail how to set up an example autodial directory.

Each time you start V-Term, it first tries to load a 'default' parameter file. It then tries to load a file called 'autodial.vtm'. This file contains the information stored in the autodial table. If you then type Alt-A Alt-D to select 'Dial' on the 'Autodial' menu, you will be prompted for a system name. V-Term searches the Autodial directory for the system which starts with what you type (you usually only need to type the first two or three letters), displays that entry, and then does the following:

- Loads the corresponding parameter file, if one is specified.
- Resets the on-screen timer to 00:00:00
- Displays V-Term's main menu
- Sends the Modem string for that system to the modem
  (The Modem string will be a modem command to dial the modem.)

For example, if you use V-Term to call Delphi, CompuServe, your friend Joe's BBS, and your office computer, your autodial table might look like:

System Name	Parameter File	Modem String
Delphi	DELPHI	ATDT555-1234<cr>
CompuServe	COMPUSRV	ATDT555-5678<cr>
Joe's BBS	JOESBBS	ATDT555-7777<cr>
Office	OFFICE	ATDT555-1976<cr>

To set up this table, go to the autodial menu, and select Item until entry number 1 is displayed. Then select Name, and at the prompt type 'Delphi <ENTER>'. Then select Filename, and at the prompt type 'delphi <ENTER>'. Finally, select ModemString, and at the prompt type 'ATDT555-1234<Ctrl-M><ENTER>'. The Ctrl-M is the character which is usually sent by <ENTER>. You can use that in any macro definition or modem string which should contain an <ENTER>. These examples assume you are using an auto-dial modem with Haye's style AT commands. If not, you should substitute the commands your modem does use. If you don't have an auto-dial modem, you can simply leave the Modem String blank, and use the Autodialer to manage the Parameter files for the different systems. Set up the remaining entries in a similar fashion. Once you have set up this table, you can call up Delphi with all parameters correctly set by simply typing: Alt-A Alt-D 'd<ENTER>'.

X. Using the Capture Menu

Starting with ver. 3.0 of V-Term, all of the functions which relate to the current capture buffer have been collected together on the Capture menu. From this menu, you can set the name of the buffer to use for Capture, turn capture on and off, Copy the current screen into the Capture buffer, print the current capture buffer, save the current capture buffer, or kill the capture buffer to remove it from memory. Most of these operations are fairly self-explanatory, so we will simply outline some different ways in which users have found these useful.

Print Screen - One way to use these features is to be able to print the screen periodically. To do this, leave Capture off, and whenever you wish to print the screen, first Kill the old capture buffer, then Copy the Screen into the capture buffer, then Print the capture buffer. If you would rather do the print just once at the end, leave Capture off, then simply Copy the Screen whenever you want to save what is on the screen, then Print the capture buffer at the end of the session.

Printer echo - Another usage of these features is to keep a printed record of your entire session. To do this, Kill the old capture buffer, and turn Capture On at the beginning of your session. You can then either select Print at the beginning to have the session printed out as you go along, or select Print once at the end to have a complete record of your session on-line. If you wish to only capture parts of your session, it is probably best to Print out the session at the end, as the print will automatically stop whenever it reaches the end of the capture file when capture is turned off. Note to 128k users: Since the entire capture buffer is stored in memory, you may want to periodically kill the capture buffer. To do this, wait until the host stops, then turn capture off, wait for the Print On message to disappear, then kill the buffer, turn capture back on, and select Print again. You should do this whenever you see the memory available get down to 8k. If V-Term runs out of memory, it will automatically turn Capture off.

XI. Disk Files

Filenames

First, the basics. For many operations, V-Term will prompt you for a filename. V-Term takes the name you type and does the following: First, it converts it to all uppercase, then, if no extension is specified, it adds a default extension. For parameter files, the default extension is .VTM. For other files, the default extension can be set with the Extension option on the Disk menu. For drive numbers, V-Term will accept filenames that look like "filename.ext:3" with drive numbers up to 255. It will also accept filenames of the format "2:filename.ext" with drive numbers 0-9. Do NOT use drive numbers not supported by your disk system. When checking for files on disk, V-Term ignores case when comparing filenames, so that "File1.doc", "FILE1.DOC", "file1.doc", and even "FilE1.DoC" are all considered the same.

Filetypes

Disk Basic uses two values to discern the type of a file on disk. V-Term classifies these according to common usage as ASCII, Binary, and Compressed BASIC files. ASCII refers to most word processor files, text files, and BASIC programs stored with the ,A option. (Note: most BBS's and information services seem to prefer that BASIC programs be uploaded in ASCII format.) Binary refers to Machine Language programs, graphics files, and program-specific data files (including some word processing files with special format commands in them). Compressed BASIC refers to the most common kind of stored BASIC program. When uploading or downloading files, or when saving/loading files, V-Term transfers Binary and Compressed BASIC files as-is. However, since many computers store ASCII files in slightly different formats, V-Term attempts to process ASCII files into a standard form for file transfer. V-Term will add or remove LF characters, as appropriate, and will sometimes remove other control characters from a file, especially during ASCII and Line protocol uploads. For this reason, whenever you are unsure what is in a file, or want to gaurantee that V-Term transfers the file EXACTLY as it appears on disk, then you should use Binary. So, you should use Compressed BASIC for compressed BASIC programs, ASCII for ordinary text and documentation, and Binary for everything else.

How to use V-Term with non-standard DOS's

V-Term was developed on a system using ADOS3 and 80 track drives, and works well in that environment. V-Term has been tested and used on a variety of other systems and DOS's with very few problems. Starting with ver. 3.0, a number of changes have been made to make V-Term even more compatible with non-standard DOS's, especially those which support various hard disk interfaces. The most important concern before using V-Term with an unusual DOS (i.e. a DOS other than RSDOS 1.0 or RSDOS 1.1) is that the new DOS be running in a ROM or EPROM. Many third party DOS's are provided on disk, and can be run by simply loading them from disk. In this environment, V-Term will NOT work correctly, and will probably not even start correctly.

To set up V-Term for use with a particular non-standard DOS, you should set the Tracks value on the Disk menu to the number of tracks used for each drive on your system. V-Term will currently only allow one setting of this parameter. If you work with a DOS that allows different-sized disk drives, then you should set this parameter to the smallest disk size that you will be writing to with V-Term. Note that this parameter has no effect on how V-Term reads files. V-Term will correctly read from disks with anywhere from 17 to 129 tracks, as long as they have the directory on track 17, and an RSDOS format GAT and directory structure. The Tracks parameter simply sets the highest-numbered track that V-Term will write files to. This parameter is saved in Parameter files, and so you should be sure to save this in your DEFAULT.VTM cofiguration file.

For those with particular problems using V-Term with a third party DOS, you should check with the author of the DOS that the DSKCON routine resides wholly between $C000 and $E000, and that all variables used by DSKCON are below $400. Any third party DOS which implements DSKCON in a manner compatible with this should work correctly with V-Term.

V-Term and RAMdisks

Since many custom DOSes now include support for RAMdisks which use the additional memory in a 512k CoCo3, we have added an option to override V-Term's normal automatic memory check. When the Mem128k option on the Buffer menu is enabled, V-Term will only use 128k in a 512k machine. This prevents V-Term from conflicting with a RAMdisk which uses that memory.

XII. Buffers

In order to make most efficient use of the available memory on the Color Computer 3, V-Term uses a ramdisk-style buffer interface which allows multiple files to be stored in memory simultaneously. (In this document, files stored in memory are often referred to as "buffers".) On a 128k computer,

For the more technically oriented:

ASCII - causes nulls and line feeds to be stripped, and stores the file as "1 A", text file, ASCII.
Binary - no processing is done, and the file is stored as "2 B", machine language, binary.
Compressed- no processing is done, and the file is stored as "0 B", Basic program, Binary.

This is more fully documented in the Disk Extended Color BASIC manual.

Why Buffers?

V-Term's buffers provide a way to store a number of unrelated files in memory simultaneously. Unlike many terminal programs, it is not necessary to download a file to memory, then save to disk before downloading another file. You can either download direct to disk, or save all of the files in memory, then save them to disk after you have logged off (which is especially nice on pay systems). V-Term's buffers are NOT disk files, however. Since buffers are stored in memory, they can be accessed much faster than even a hard disk. Using buffers can result in faster file transfers, and smoother operation of the program. Accessing buffers does not require turning off interrupts, so no characters are lost, and full typeahead can be preserved. This allows V-Term to list buffers to the screen or printer even during downloads. In the terminology that OS9 users have come to appreciate, buffer access in V-Term is 'no-halt'.

Special Notes

There are a few peculiarities of V-Term's buffer system which more advanced users may want to be aware of. First, the Capture buffer is the same as any other buffer, and can be manipulated using the options on the buffer menu. The Capture menu options are simply shortcuts to doing the same thing. Secondly, adding LFs to ASCII files when loading them into a buffer, while still supported in ver. 3.0, is largely unnecessary due to the ASCII processing done automatically during uploads.

Finally, starting with ver. 3.0, V-Term has a somewhat unusual way of dealing with filenames when saving or loading buffers. When saving a buffer to disk, for example, you are prompted first for the buffer name, then for the disk file name. In order to minimize typing, since usually the two names will be similar, if not identical, whatever name is typed for the first will be 'pre-typed' for the second. If you don't like this name, Shift-<back arrow> will erase that name so you can type another one. The name can also be edited with backspace just as if you had typed it yourself.

XIII. File Transfers in V-Term

What is a File Transfer Protocol?

There are two common uses of computer communications today. The first is the exchanging of messages or EMail. The other is to exchange programs or datafiles. For the first, we usually just type in our message or read it from the screen. For the second, we need to have some way to have the terminal program read the file from disk or memory and send it to the other computer. A file transfer protocol is a set of rules which can be used to send a file from one computer to another. The full description of these rules can get fairly complicated, but you don't need to know how they work. You simply need to know what they are used for and how to use them.

V-Term uses two basic file transfer protocols. XModem is a very old protocol which can be used to transfer any type of file between any two computers (assuming of course, that they both can use the XModem protocol). V-Term can either send or receive files using XModem. V-Term also supports XModem-CRC, and YModem (more correctly referred to as XModem-1k), which are derivatives of basic XModem. V-Term also supports two methods specifically designed for uploading ASCII files. Each has some provision for clean communication built in. The "Line oriented ASCII transfer protocol," or simply the Line protocol, allows you to easily compose messages off-line, using a text editor or word processor, and then send them to a BBS or Information Service as if you had typed them in directly. It does this by waiting for a specific prompt character and then sending each successive line only after the prompt character appears. The ASCII transfer is similar, but waits only for a pause from the host before sending the next line. Note that the Line protocol is similar to the Prompted ASCII protocol implemented on many other terminals.

XModem, XModem-CRC, and YModem Protocol File Transfers

XModem (sometimes referred to as Christensen's protocol, or Modem7 protocol) is a popular file transfer protocol used by many microcomputers and Bulletin Board Services. Originally developed by Ward Christensen for use on CP/M systems, this is probably the single most widely used file transfer protocol around. Over the years, a number of improvements and extensions to XModem have been proposed and used. One of the most common is to substitute a 16-bit Cyclic Redundancy Check (CRC) for the 8-bit checksum used for error detection in standard XModem. This is referred to as XModem-CRC. YModem, also known as XModem-1k, adds the ability to send longer packets than XModem. A single YModem packet can be either 128 bytes or 1024 bytes. For connections over networks, YModem's larger packet size usually makes it significantly faster than XModem. However, on noisy connections, YModem can actually be slower, due to the extra overhead of re-transmitting long packets. For more complete technical details, refer to Appendix C. In the rest of this discussion, we will use 'XModem' to refer to any of these three protocols.

In order to receive a file from a remote host using the XModem protocol, first select 'XModem' using the Protocol option on the Transfer menu, then select the device on which to store the incoming file (either memory or disk), and finally, select the filetype if you will be saving it to disk. (If in doubt, use Binary, you can change it later, by loading it back into V-Term and re-saving it. See the chapter on Buffers.) After the other computer has started the XModem transfer, you can start V-Term's file transfer. When you select Receive on the Transfer menu, you will be prompted for a filename, and then for a filetype. Simply type 'A' for ASCII files, 'B' for binary files, or 'C' for Compressed BASIC files. The protocol transfer will begin as soon as you select the filetype. A status line will appear on the screen to keep you informed of the progress of the transfer, and the final status will be echoed to the alternate text screen when the transfer is complete. If you select ASCII as the filetype, then V-Term automatically converts CR/LF pairs in the incoming file into CR (and converts single LF characters into CR characters). It also removes Nulls and Ctrl-Z's from the incoming file.

To send a file, after the other computer is ready to receive the file, select the device and select Transmit. After typing the filename and pressing Enter, you will be asked if this is an ASCII transfer. If you answer 'Y', then V-Term will automatically add a LF after ever CR in the file, in accordance with the standard for transferring ASCII files with XModem. It will also filter out nulls and certain other control characters from the file.

Be careful not to select Receive or Transmit until you are ready. If you are working with a remote host, prepare it for the file transfer first, then start V-Term's file transfer. Note that once you select Receive or Transmit, V-Term will not allow you to type to the remote host until the transfer is complete. This is to prevent accidentally causing errors in the transfer. You can, however, abort the Receive or Transmit by pressing Break before you finish entering the filename, or by selecting Abort once the transfer has begun. You may have to type Control-X several times to stop the remote host after selecting Abort. (Some hosts will recognize Ctrl-C's as stop characters, and for some, you will need to press Ctrl-U repeatedly until the host errors out.)

Line and ASCII protocol file transfers

These protocols have been designed to speed the sending of ASCII text files to a host by allowing V-Term to "automatically type" the file into a text editor. In most line-oriented text editors, there is some sort of prompt at the beginning of each line to be typed in. V-Term handles Line protocol transfers by waiting for the appropriate prompt character, sending one line of the ASCII file, and then repeating this process for each line of the file. This is especially useful for use in EMail or message systems.

The ASCII protocol is similar in that it processes the text to be sent as individual lines. However, it does not wait for a prompt, but instead simply waits for a pause from the host before continuing. The length of pause is settable when you start the transfer. After typing in the filename, you are asked for a value for the delay. The value in parentheses is the default value that will be used if you simply press <ENTER>. This value is in

Now, to send this message to Sarah on your local BBS, the process might look something like the following:

```
Hypothetical BBS Ver. 1.0
Message Menu
   <R>ead Messages
   <S>end a message
   <E>xit to Main Menu
What do you want to do? S <ENTER>

Who is the message to? SarahJones <ENTER>
What is the Subject? New Club Idea <ENTER>

To: SarahJones
From: CorinneSmith
Re: New Club Idea
Is this correct (Y/N) ? Y <ENTER>

Type your message below.
Type Control-Z on a line by itself to end.

1:
```

[At this point, we are in the editor, ready to type in the text. The prompt character (the character that will be displayed before every line) is a colon, so we go to the Transfer menu, select Disk under the Device option, and Line under the Protocol option, then select Transmit. We are asked for the filename, and type 'SARAH.TXT<ENTER>'. We are then asked for the prompt character, and type <:><ENTER>. Now V-Term will type the message, one line at a time, until it gets to the end. (the '1:' won't be repeated on your screen, we'll repeat it here for clarity)]

```
1:Sarah,
2:       I think your idea for a new computer
3:club in this area specifically for the
4:Color Computer III is a great idea!
5:Especially with the large number of new
6:Color Computer III owners.  Since the old
7:Color Computer I/II club is doing so well,
8:it won't hurt them if we start a separate
9:club.  We would want to be careful to
10:maintain close relations with them, though.
11:Any ideas on a good name?
12:                          Corinne
13:
```

[Now, V-Term is finished with the file transfer and stops. Since every system uses it's own unique way to indicate when you're done typing the message, it can't be built in to V-Term. So we press Ctrl-Z, according to the instructions from the BBS, to finish. (again, we repeat the '13:' for clarity)]

13:^Z
<E>dit, <S>ave, <Q>uit, or <C>ontinue? S <ENTER>

Every Bulletin Board System will have it's own peculiarities. This is merely an example that is typical of many BBS's. The only major difference that you may see is that there are a few BBS's which do not show a prompt before each line. On these systems, simply press <RETURN> when asked for a prompt character, and the transfer should work correctly. For more detailed information, see Appendix B.

XIV. Input in V-Term

Most of the parameters and options in V-Term can be adjusted merely by selecting a menu item. Some, such as the Color setting on the Terminal Menu, may require selecting the menu item several times to cycle through the available options. Using V-Term's key repeat feature can help such selections greatly, and using Parameter files should keep the user from needing to adjust such parameters more than once.

However, some things such as filenames cannot be easily selected from a menu. For such options, V-Term uses the bottom menu line to prompt the user to type a response. When such a prompt appears, all keys typed will be echoed to this input line. Input is terminated with <ENTER>, and can be aborted at any time by pressing <BREAK>. A few such input prompts which only require a single character for input will only wait for a single character, and do not require that you press <ENTER>.

A few keys act slightly differently when used in the input line than they do at other times. Regardless of any mode settings, the back arrow and Clear key both act as backspaces. Also, any control character except Null can be included by simply typing the character. This is useful in Macro definitions and Modemstrings, particularly the ability to include <ENTER> in the definition by typing Ctrl-M. If you are using the Conference Mode to type commands to a host for which the echo makes backspace editing tedious, you can send single control characters to the host by typing the control character and then <ENTER>. If the <ENTER> is immediately preceded by a control character, it will not be sent.

XV. Graphics and Hardware text displays

V-Term supports two different 80-column text displays. One is a standard 80-column by 28 line text screen generated by the GIME, and the other is a 640 x 225 4-color graphics screen, which V-Term uses to display a variety of character sizes and styles that are not available on the hardware display. Each of these displays has certain advantages and disadvantages. Each of V-Term's two screens can be set independently to use either one with either the Ctrl-F1 command key, or with the Display option on the Terminal menu.

Memory: The graphics screen uses 32k more memory than the hardware screen. Notice that if you are using the hardware screen and fill up buffer memory, you will sometimes see the message "Graphics memory not available!" when you try to switch to the graphics display. Since V-Term releases the graphics memory for use by buffers when using the hardware display, it is sometimes impossible to switch to the graphics display because that memory is in use by buffers. Note that the extra memory is only made available for buffers if BOTH screens are set to use the hardware display.

Flexibility: The graphics screen can display double size characters, and will correctly display the special line-drawing characters and other special characters which are not available on the hardware screen.

Speed: The hardware screen is significantly faster than the graphics screen.

Legibility: The heavier vertical lines in the font used on the graphics screen make it significantly more legible, especially on poor quality monitors. The font used can even be quite readable on a color TV set, and V-Term should be usable with color TVs that do not suffer from overscan problems.

XVI. Conference Mode

There are several situations in which it is convenient for the terminal to be 'smart' in the sense that it should accept a full line of input from the user, then send that full line to the host. Such a facility is often referred to as a 'Conference' mode, since it is commonly used in multi-user conferencing systems, such as those available on CompuServe and Delphi. One common annoyance with such systems is that messages from other users can often appear in the middle of the line you are in the process of composing. V-Term offers a Conference mode which is accessible through the Input option on the main menu. 'Normal' Input means that each character you type is immediately sent to the host. 'Conference' Input means that an input line appears into which you type whatever you desire, and the entire line is sent to the host when you press <ENTER>. As long as 'Conference' Input is enabled, this input line will appear whenever another input line is not on top of it. Since this mode is stored in the Parameter file, you can configure V-Term so that you will be in Conference mode whenever you use V-Term with a particular host. Refer to the section "Input in V-Term" for more information on including control characters within the Input line.

The Conference mode is also useful whenever you are using a system across a network. The author, for example, often uses it on Delphi, since it is much faster to backspace and edit commands in Conference mode than to wait for the echo from Delphi.

XVII. If You Have Problems

Read the Manual

Silly as it may sound, if you're having a problem getting V-Term to do something, or if you can't understand why it's doing something, you should first make sure you read the relevant parts of this manual. We've tried to make this manual as complete and accurate as possible, so the information you need should be in here. Also, if V-Term is not behaving correctly on some particular system, it may be that V-Term is not correctly configured. Get your SysOp, System Administrator, or some other knowledgable person to help you if you can't understand how V-Term should be configured. If none of this helps, then try writing us with a description of your problem, and we'll try to help you. Remember, though, that we can't help you with many things since we simply don't know every computer system that you may try to use V-Term with.

Bugs and Other Nasty Critters

If you've tried everything, and still can't get V-Term to act as you think it should, then you may have found a bug. We've tried to make V-Term as bug-free as possible, but since we're only human, it seems inevitable that some bugs will get by us. Hopefully, you will never find one, but if you discover a problem that you believe is a bug in V-Term, we would appreciate knowing about it so that it can be fixed in later releases.

If you believe you've found a bug, please write us with the following information:

- A description of your computer system
  (A list of all relevant hardware, including model numbers and manufacturers where appropriate)
- The DOS that you are using
  (e.g. RSDOS ver. 1.0, ADOS 3, etc., including any custom modifications that you've made)
- The version of V-Term that you are using
  (copied from the 'Ver' option on the Terminal menu. It is of the form v.r.n. where v is the version, r the release, and n the number of the release.)
- The settings of V-Term that you are using
  (RS232 settings, terminal mode settings, and printer settings if appropriate)
- The system that you were communicating with
  (e.g. Delphi, CompuServe, School VAX, local CoBBS, etc.)
- The program that you were using on the remote system
  (e.g. emacs editor, full-screen mail, conference system, etc.)
- Your terminal configuration on the other computer
  (e.g. VT100, local BBS doesn't support different terminals)
- What you were trying to do
- How you tried to do it
- What you expected to happen
- What actually happened

- Were you able to repeat it?
  (Some errors can be caused by noise, or occur due to complicated circumstances that can be difficult to find. If you were able to cause the same thing to happen again, then there is a better chance that we can cause the same error to occur, which helps us in diagnosing it.)
- Your name, address, and telephone number, so that we can contact you if we need additional information.
  (If we can contact you through CompuServe, Delphi, Bitnet, or another electronic mail system, then we may be able to get back to you more quickly.)

XVIII. Appendices

Appendix A: Technical description of terminal emulations

This section documents the control codes and escape sequences that V-Term responds to in each of it's emulations.

Part 1: VT100 emulation

For further information, see the VT100 User Manual, especially Chapter 3: Programmer Information.

Control Codes

Hex	Dec	Name	Function
00	0	Null	No action
07	7	Bell	Causes tone from monitor.
08	8	BS	Causes cursor to backspace, unless at top left corner.
09	9	HT	Move cursor to next tab stop, or to right margin.
0A	10	LF	Causes a Line Feed or new line operation, according to the Line Feed/New Line Mode
0B	11	VT	Treated as LF.
0C	12	FF	Treated as LF.
0D	13	CR	Move cursor to left margin on current line. V-Term: see Send CR mode
0E	14	SO	Invoke G1 character set.
0F	15	SI	Invoke G0 character set.
18	24	CAN	Causes ESC sequence to be cancelled
1A	26	SUB	Interpreted as CAN
1B	27	ESC	Introduces an escape sequence.
7F	127	DEL	Ignored

Escape Sequences

Cursor Movement Commands

Cursor Up	ESC [Pn A	
Cursor Down	ESC [Pn B	
Cursor Forward	ESC [Pn C	
Cursor Backward	ESC [Pn D	
Direct Cursor Addressing	ESC [Pl ; Pc H	or
	ESC [Pl ; Pc f	

Index	ESC D
New Line	ESC E
Reverse Index	ESC M
Save Cursor Pos'n	ESC 7
Restore Cursor Pos'n	ESC 8

Note: Pn refers to a numeric parameter, which consists of a sequence of ASCII digits, with optional leading zeros, which express the decimal value of the parameter. If omitted, or specified as "0", then an appropriate default value is used.

Example: ESC [015 B will move the cursor down 15 lines; ESC [A will move the cursor up one line (default is one.); ESC [0 ; 000 H will move the cursor to the the upper left hand corner.(both parameters default to one.)

Pl, Pc refer to the line and column, respectively. Line and column numbers start at 1. If Origin Mode is enabled, line numbering starts at the first line of the current scroll region, otherwise, it starts at the first line of the screen.

Character Attributes

ESC [Ps ; Ps ; ... ; Ps m

Ps is one of the following options, which are processed in the order they appear:

0 or no options	All attributes off
1	Bold characters on
4	Underline on
5	Blinking characters on
7	Reverse video on

Any other parameter values are ignored.

Note: Blinking, Bold and underline attributes are displayed, but not stored, they will not appear in screen Snapshots. Also note that Blinking characters are only available in the Hardware text screen version. Bold characters are displayed in the menu colors.

Erasing

From Cursor to End of Line	ESC [K or ESC [0 K
From beginning of line to cursor	ESC [1 K
Entire line containing cursor	ESC [2 K
From cursor to end of screen	ESC [J or ESC [0 J
From beginning of screen to cursor	ESC [1 J
Entire Screen	ESC [2 J

Character Size

These codes affect the entire current line. Note that only single width/single height characters are available on the hardware screen.

Double Height/Double Width, top half	ESC # 3
Double Height/Double Width, bottom half	ESC # 4
Single Width/Single Height	ESC # 5
Double Width/Single Height	ESC # 6

Double Width characters reduce the number of characters per line by half. Using double height characters requires that two lines be used, one to display the top half, one for the bottom half, and that the same information be sent to both lines.

Character Sets

Character Set	G0 designator	G1 designator
United Kingdom (UK)	ESC (A	ESC) A
United States (USASCII)	ESC (B	ESC) B
Special graphics	ESC (C	ESC) C

Note: Special graphics characters will not display correctly in The hardware text screen version. After setting the character set with one of these sequences, it may be selected for display with SO or SI, see Control Codes, above.

Scrolling Region

ESC [Pt ; Pb r

Pt is the number of the top line of the scrolling region; Pb is the number of the bottom line. Pb must be greater than Pt. This command moves the cursor to the current origin, as defined by the setting of the Origin Mode. Default values are Pt = 1, Pb = 24.

Tab Stops

Set Tab at current column	ESC H
Clear Tab at current column	ESC [g or ESC [0 g
Clear all tabs	ESC [3 g

Modes

To set a mode, use ESC [Ps h
To reset a mode, use ESC [Ps l (lowercase L)
Where Ps is one of:

Ps	Mode Name	Set state	Reset State
20	Line Feed/New Line	New Line	Line Feed
?1	Cursor Key Mode	Application	Cursor
?2	ANSI/VT52 Mode	ANSI (VT100)	VT52
?5	Screen Colors	Reverse	Normal
?6	Origin Mode	Relative	Absolute
?7	AutoWrap	On	Off
?8	AutoRepeat	On	Off

Set Keypad Application Mode ESC =
Reset Keypad Application Mode ESC >

Reports

Cursor position report
  Invoked by ESC [6 n
  Response is ESC [Pl ; Pc R
Status Report
  Invoked by ESC [5 n
  Response is ESC [0 n (terminal OK)
What Are You
  Invoked by ESC [c or ESC [0 c
  Response is ESC [?1 ; 0 c
  * Alternately invoked by ESC Z. (Not recommended.)
Report Terminal Parameters
  Invoked by ESC [0 x or ESC [1 x
  Response is ESC [3 ;<par>;<nbits>;<baud>;<baud>;1;0 x
    Where<par> is either 1 : no parity,
                         4 : odd parity, or
                         5 : even parity;
      <nbits> is either 1 : 8 bits/char, or
                        2 : 7 bits/char; and
      <baud> is either 48 : 300 or 450 baud,
                       56 : 600 baud,
                       64 : 1200 baud,
                       88 : 2400 baud,
                      104 : 4800 baud, or
                      112 : 9600 baud.

Miscellaneous

Screen alignment display ESC # 8
  Fills screen with uppercase 'E's.
Terminal reset ESC c
  Identical to Alt-Clear, this clears the screen, and resets the character set, tab stops, and a few other minor settings.
Confidence test ESC [2 ; Ps y
  On the VT100, this invokes a self-test routine which depends on the value of Ps. In V-Term, Ps is ignored, and this is identical to ESC c.

Keyboard interpretation

All control sequences and standard graphics characters have the codes defined in the ANSI standard x3.4-1977 (ASCII standard).

The remaining keyboard keys send escape sequences as defined below, depending upon the states of the ANSI mode (ANSI) and the Cursor Key mode (CKM), described above.

Arrow Key	VT52 mode	ANSI set, CKM reset	ANSI, CKM set
Up	ESC A	ESC [A	ESC O A
Down	ESC B	ESC [B	ESC O B
Right	ESC C	ESC [C	ESC O C
Left	ESC D	ESC [D	ESC O D

Auxiliary Keypad Codes (See section VII for additional info)

These codes are returned if the Keypad Application Mode is Set.

Key	VT52 mode	ANSI mode	Key	VT52 mode	ANSI mode
F2-0	ESC ? p	ESC O p	F2-(-)	ESC ? m	ESC O m
F2-1	ESC ? q	ESC O q	F2-(,)	ESC ? l	ESC O l (*)
F2-2	ESC ? r	ESC O r	F2-(.)	ESC ? n	ESC O n
F2-3	ESC ? s	ESC O s			
F2-4	ESC ? t	ESC O t	F2-ENTER	ESC ? M	ESC O M
F2-5	ESC ? u	ESC O u	PF1	ESC P	ESC O P
F2-6	ESC ? v	ESC O v	PF2	ESC Q	ESC O Q
F2-7	ESC ? w	ESC O w	PF3	ESC R	ESC O R
F2-8	ESC ? x	ESC O x	PF4	ESC S	ESC O S
F2-9	ESC ? y	ESC O y			

(*) The final character in these escape sequences is a lowercase 'L'.

If the Keypad Application mode is Reset, then the F2 combinations listed above return just the printed character, i.e. F2-6 returns '6'.

Part 2: VT52 emulation

Control codes (Same as VT100)

Escape Sequences

Cursor Up	ESC A
Cursor Down	ESC B
Cursor Right	ESC C
Cursol Left	ESC D
Select Special Graphics	ESC F
Select ASCII characters	ESC G
Cursor to home	ESC H
Reverse line feed	ESC I
Erase to end of screen	ESC J
Erase to end of line	ESC K
Direct cursor address	ESC Y l c ("l" is CHR$(line number+31)) ("c" is CHR$(column number+31)) (line and column numbers start at 1)
Identify	ESC Z (response is ESC / Z)
Set Keypad Application Mode	ESC =
Reset Keypad Application Mode	ESC >
Enter ANSI (VT100) mode	ESC <

Keyboard Interpretation (See VT100 description)

Part 3: VIDTEX emulation

Control codes

Hex	Dec	Name	Function

Part 4: "CRT" emulation

Control codes

Hex	Dec	Name	Function
8	8	BS	Backspace
9	9	HT	Tab
A	10	LF	Line Feed
B	11	VT	Reverse Line Feed
C	12	FF	Erase screen/home cursor
D	13	CR	Carriage Return/move cursor to left margin

Escape sequences (None.)

Keyboard Interpretation

Arrow keys are interpreted with Arrow Key mode considered Off, PF1-PF4 are ignored, and the other terminal emulation keys are interpreted as if Keypad Application Mode is permanently reset.

Arrow key	Control Code
Left	8, BS
Right	9, HT
Down	10, LF
Up	11, VT

Appendix B: Description of Line protocol

V-Term's Line protocol includes a number of safeguards to help ensure that the relatively simple model used to drive the transfer will work. A more careful description of the protocol is included here so that users can diagnose problems, or can use this feature in certain unusual situations.

Flow control

The Line protocol will stop transmitting whenever it receives an XOFF character, and will resume when it receives an XON character.

Line oriented file transfer

Except for the subtleties noted above, the transfer is straightforward. V-Term sends the first line without waiting for any prompt, and then repeats the following sequence until the end of file is reached:

- Wait for a CR character.
- Wait for the prompt character, if any.
- Fetch the next line from disk or memory
- Send the line followed by a CR
  (LF characters and nulls are not sent)

Using the Line transfer

The Line protocol can be used any time a series of prompts which share a common character need to be answered. One application of this is in the programming of certain modems, where the modem always responds with a '>' prompt.

One unusual situation occurs if there is no prompt character. In this case, you can try either a null prompt character, or a LF prompt character.

ASCII Protocol

The ASCII Protocol is identical to the Line Protocol in operation, except that it does not wait for a CR or prompt character. Instead, it waits for a pause from the host before fetching the next line. The length of this pause is measured in 1/60 second units, and is settable at a prompt after you select 'Transmit'.

Appendix C: Description of XModem protocol

The XModem protocol was originally developed by Ward Christensen for transferring binary files between CP/M computer systems. Since then, it has become one of the most popular protocols for file transfers involving microcomputers. This is primarily because of the relative simplicity of the protocol, and because it does provide error detection and correction capabilities.

Five special characters are used with the following significance:

SOH - Start Of Header - marks the start of a 128-byte packet ($01, or Ctrl-A)
STX - Start of Transmission - marks the start of a 1024-byte packet ($02, or Ctrl-B)
EOT - End Of Transmission - marks the end of the file transfer ($04, or Ctrl-D)
ACK - ACKnowledge - used to indicate that the last operation was completed successfully ($06, or Ctrl-F)
NAK - Negative AcKnowledge - used to indicate that the last operation was not completed successfully ($15, or Ctrl-U)
CAN - CANcel - used to abort the file transfer. V-Term will abort if it receives two consecutive CAN characters. The Abort option for XModem will send 5 CAN characters followed by 5 BS characters to erase the CANs from the line input buffer on the host. ($18, or Ctrl-X)

Each packet looks like:

1-byte	SOH character or STX character
1-byte	block number
1-byte	one's complement of block number
128-bytes	data (1024 bytes if first character is STX)
1-byte	checksum - sum of the 128 data bytes (2-byte CRC for CRC error detection)

The last packet is traditionally filled with Ctrl-Z's.

The entire transfer is receiver-driven, meaning that the sender does nothing until prompted by the receiver.

1) There is an initial handshake to determine the error detection method to use The receiver sends an NAK to request checksum error detection, or a 'C' (ASCII $43) to request CRC error detection. In the case of XModem-CRC or YModem, the receiver sends 'C' at 10-second intervals until the transmitter responds by sending the first packet. If the receiver sees no response after 5 tries, it switches to checksum mode.

2) The sender acknowledges receipt of the handshake character by sending the first packet.

3) The receiver checks the block number and the check value. If they are correct, the receiver prompts for the next packet with ACK, otherwise, it prompts for a re-transmit of the current block with NAK.

4) If necessary, the transmitter re-sends the packet.

5) Steps 2, 3, and 4 are repeated until the file is exhausted.

6) After the complete file is successfully sent, the transmitter sends an EOT character to the receiver, and the receiver sends an ACK back.

There are a number of optimizations to this basic method which can reduce the chance of errors, and improve the speed at which errors are recovered from. Several are outlined in a document by Chuck Forsberg of Omen Technology, the author of the YModem protocol.

Appendix D: Parameter File description

Since a large amount of very critical data is stored in the parameter files, you should not attempt to alter these files directly. An error check code is included to help guard against corrupted parameter files. The following list gives the information stored in the Parameter file as a reference only.

- A version and copyright notice.
- Current screen format
- Disk parameters, number of tracks, default drive, default extension
- Palette values
- Transfer Device, Protocol, and Filetype
- Current emulation
- Terminal Modes
- Saved cursor position
- G0 and G1 character sets
- Printer variables
- RS232 status
- Buffer Capture Name
- Capture status
- Tab settings
- Shift-lock status values
- Timeout value for ASCII protocol
- Conference mode status