

Novaterm

version 9.6

User's Guide

Copyright © 1989-1996 by Nick Rossi

The Author of this software and manual is Nick Rossi.

The Author of this software makes no warranty of any kind, either express or implied, including but not limited to implied warranties of merchantability and fitness for a particular purpose, with respect to this software and accompanying documentation.

In no event shall the author be liable for any damages (including damages for loss of business profits, business interruption, loss of business information, or other pecuniary loss) arising out of the use of or inability to use this program, even if the author has been advised of the possibility of such damages.

All rights reserved. No part of this document may be reproduced, in any form or by any means either manually or electronically, without written permission from the Author.

TABLE OF CONTENTS

Introduction	i
About the software	i.1
FTP site.....	i.2
Special thanks	i.3
Getting started	1
Starting up the program.....	1.1
Selecting a modem type.....	1.1.1
Specifying your modem's maximum speed	1.1.2
Selecting a serial device	1.1.3
Selecting a dialing method	1.1.4
Selecting a C128 VDC screen driver.....	1.1.5
Specifying colors and fonts.....	1.1.6
Specifying device settings.....	1.1.7
Specifying the memory device.....	1.1.8
Dealing with a hardware conflict.....	1.1.9
Selecting a real-time clock device	1.1.10
Novaterm basics.....	1.2
Using menus	1.2.1
Entering text.....	1.2.2
The modular approach	1.2.3
Getting familiar with Novaterm	2
The Main Menu: a quick tour	2.1
Terminal mode.....	2.1.1
Dial a number	2.1.2
Configuration.....	2.1.3
Disk operations	2.1.4
Buffer menu.....	2.1.5
Device settings.....	2.1.6
Utility modules	2.1.7
Exit terminal	2.1.8
On-line activities	3
Terminal mode.....	3.1
Terminal emulations	3.1.1
Status line.....	3.1.2
Commodore key commands	3.1.3
Function keys.....	3.1.4
Key reassignments	3.1.5
Notes on 80 column mode	3.1.6
Improving readability	3.1.6.1

Compensating for slow display.....	3.1.6.2
Displaying color	3.1.6.3
Dialing directory	3.2
Creating a dialing entry.....	3.2.1
Directory commands.....	3.2.2
Dialing a number	3.2.3
Multiple dialing.....	3.2.4
Using the buffer	3.3
Specifying the memory device.....	3.3.1
The buffer directory.....	3.3.2
Working with buffer subdirectories.....	3.3.3
Capturing text in the buffer.....	3.3.4
Downloading a file to the buffer	3.3.5
Buffer file options.....	3.3.6
Buffer directory	3.3.6.1
Disk directory	3.3.6.2
Change buffer device.....	3.3.6.3
Load files from disk.....	3.3.6.4
Save files to disk.....	3.3.6.5
Rename file.....	3.3.6.6
Scratch file.....	3.3.6.7
Clear subdirectory.....	3.3.6.8
Buffer file processing options.....	3.3.7
View file.....	3.3.7.1
View file in 80 columns.....	3.3.7.2
Print file.....	3.3.7.3
File to modem.....	3.3.7.4
ASCII to PET	3.3.7.5
PET to ASCII	3.3.7.6
PET to Unix.....	3.3.7.7
UUdecode file	3.3.7.8
UUencode file.....	3.3.7.9
UUencode in PET.....	3.3.7.10
Editing a file in the buffer.....	3.3.8
Buffer recovery feature	3.3.9
File transfers.....	3.4
Getting files from an on-line service	3.4.1
Transferring files with a friend.....	3.4.2
Selecting a protocol.....	3.4.3
Zmodem	3.4.3.1
Ymodem batch	3.4.3.2
Ymodem-g.....	3.4.3.3
Xmodem-1k.....	3.4.3.4
Xmodem-1k-g	3.4.3.5
Xmodem-CRC.....	3.4.3.6

Punter (C1)	3.4.3.7
Multi-Punter	3.4.3.8
Kermit	3.4.3.9
WXmodem	3.4.3.10
Selecting a block size	3.4.4
Receiving a file	3.4.5
Sending a file	3.4.6
Sending and receiving summarized	3.4.7
Transferring text files from non-Commodore systems	3.4.8
Transferring files between two computers	3.4.9

Configuration **4**

Communication options	4.1
Terminal emulation	4.1.1
Baud rate	4.1.2
Word/parity	4.1.3
40/80-column settings	4.2
80-column driver	4.2.1
40-column font	4.2.2
40-column color	4.2.3
80-column color	4.2.4
Modem setting options	4.3
Modem type	4.3.1
Modem init	4.3.2
Dial method	4.3.3
Redial time	4.3.4
Dial pacing	4.3.5
Carrier type	4.3.6
File transfer options	4.4
Protocol	4.4.1
Block size	4.4.2
Chop X/Ymodem padding	4.4.3
Zmodem auto-download	4.4.4
Allow Zmodem streaming	4.4.5
Existing files policy	4.4.6
Configuration file options	4.5
Load configuration	4.5.1
Save configuration	4.5.2
Serial device settings	4.6
Serial driver	4.6.1
Flow control	4.6.2
Flow tolerance	4.6.3
XOFF character	4.6.4
XON character	4.6.5
Line pacing	4.6.6

Miscellaneous options.....	4.7
Define F-keys.....	4.7.1
Device settings.....	4.7.2
Device setting categories.....	4.7.2.1
Associated disk commands.....	4.7.2.2
Printer device.....	4.7.2.3
Phone book.....	4.7.3
F-key file.....	4.7.4
Startup script.....	4.7.5
Memory device.....	4.7.6
Set the time.....	4.7.7
Print in ASCII.....	4.7.8
Print with linefeeds.....	4.7.9
Save buffer when full.....	4.7.10
Notes on changing hardware and drivers.....	4.8
<i>Disk utilities</i>	5
Disk command.....	5.1
Disk directory.....	5.2
Textfile options.....	5.3
View textfile.....	5.3.1
View textfile in <u>80 columns</u>	5.3.2
Print textfile.....	5.3.3
Textfile to modem.....	5.3.4
ASCII file to PET.....	5.3.5
PET file to ASCII.....	5.3.6
PET file to Unix.....	5.3.7
UUdecode file.....	5.3.8
UUencode file.....	5.3.9
UUencode file in PET.....	5.3.10
Multiple file scratch.....	5.4
<i>Scripts</i>	6
Creating a sample script.....	6.1
Determining responses.....	6.1.1
Typing in the script.....	6.1.2
Executing the script.....	6.1.3
A slight variation.....	6.1.4
Starting Novaterm with the script.....	6.1.5
Script commands by category.....	6.2
Modem and screen input/output.....	6.2.1
Dialing.....	6.2.2
Communication parameters.....	6.2.3
Buffer commands.....	6.2.4
Installing drivers.....	6.2.5

Disk operations	6.2.6
Script program control	6.2.7
Time commands.....	6.2.8
Utility modules.....	7
ASCII table editor	7.1
Font editor.....	7.2
Configuration	7.3
Text editor.....	App A
Status line.....	A.1
Loading a file.....	A.2
Entering text.....	A.3
Deleting lines	A.4
Finding and replacing text.....	A.5
Changing margins	A.6
Formatting text.....	A.7
Moving, copying, and deleting text.....	A.8
Saving a file	A.9
Compiling a script.....	A.10
Other commands	A.11
Other DOS commands	A.12
BBS mode	App B
Configuration file.....	B.1
Initial setup.....	B.2
Waiting for a call.....	B.3
Logging in.....	B.4
Vendor Information.....	App C
CommPort, BBGRam, RAMDrive	C.1
SwiftLink, RAMLink	C.2
HART cartridge.....	C.3
Enhanced user port (UP9600)	C.4

INTRODUCTION

Novaterm is the best communications software available for the Commodore 64. It supports today's high-speed modems as well as older Commodore modems; it supports 80-column emulations as well as traditional Commodore color/graphics; it supports a variety of memory expansion devices for capturing text; and it supports all of the most common file transfer protocols, including Zmodem.

Novaterm is perfect for the Internet, for major on-line services, and for local Commodore and non-Commodore bulletin boards. Novaterm provides for all of your telecommunication needs!

i.1 Software support

Please feel free to write with comments and bug reports. Use electronic mail if possible! I do not advise sending postal mail, as I cannot promise a quick reply that way. Send electronic mail to one of the following addresses:

Internet: voyager@eskimo.com

GENie: N.ROSSI

Delphi: NICKROSSI

To send a comment through regular mail:

Nick Rossi
10002 Aurora Ave. N. #3353
Seattle, WA 98133
U.S.A.

i.2 World Wide Web site and FTP site

If you already have an Internet account, you can access Novaterm's web site and FTP site. The page contains a searchable on-line version of this manual among other information. The FTP site contains the older shareware version 9.5 of Novaterm, which is an excellent preview of Novaterm's abilities.

Web: <http://www.eskimo.com/~voyager/novaterm.html>

FTP: <ftp://ftp.eskimo.com/u/v/voyager/Novaterm>

i.3 Special thanks!

The following people deserve a *huge* thank-you for patiently testing the software and making valuable suggestions over several months:

Novaterm 9.6 User's Guide

Edward Piecewicz
Nate Dannenberg
Gaelyne Moranec
Rod Gasson

Tim Phelps
Gunther Richter
Daniel Dallmann
Peter Fiset

Thanks to Peter Fiset of Performance Peripherals, Inc. for assistance with the RAMDrive driver.

Thanks to Doug Cotton and Mark Fellows of Creative Micro Designs, Inc. for assistance with the RAMLink driver.

Thanks to Daniel Dallmann for writing the enhanced user port driver (UP9600).

Thanks to George Hug for writing those excellent RS-232 routines and to Transactor Magazine (RIP) for publishing them.

Thanks to Daniel Fandrich for writing and releasing a framework for the Zmodem protocol. (Mr. Fandrich's original protocol was download-only, and output data to the screen. His code laid the groundwork for completing the protocol.)

Thanks to Kermit Woodall for his WXmodem protocol source code.

Thanks to Gunther Richter for translating the software and manual into German.

And thanks to everyone who took the time to report bugs, make suggestions, and register the software. Every bit has helped to make this software successful.

1. GETTING STARTED

This section shows you how to start up Novaterm, goes through the initial configuration, and explains various basic concepts.

1.1 Starting up the program

If you are starting Novaterm for the first time, the program will help you configure your copy of the software by leading you through a series of questions. In any menu, use the cursor keys to move the highlight bar, and press RETURN to make a selection.

1.1.1 Selecting a modem type

The first item in the configuration asks for your modem type. If you are not using a 1650, 1660, or 1670 (old-style Commodore modems), *PH*
X you must select Hayes compatible as your modem type. Most consumer-grade modems today are Hayes-compatible. Other modem types are not supported at this time.

1.1.2 Specifying your modem's maximum speed

If you selected Hayes compatible as your modem, you will be asked to specify the maximum speed of your modem.

1.1.3 Selecting a serial device

Next, the configuration program asks you to specify a serial device. Novaterm uses a device-independent interface to implement serial communications. A "serial device", as described here, is any set of hardware connected to the Commodore 64 that provides an interface to a serial port. In general, the serial port itself (usually a 9-pin or 25-pin male RS232 connector) is also a part of this hardware in some way, so the serial device is the bridge between the computer and the port.

For information on where to obtain these devices, see Appendix C, **Vendor information**. Novaterm currently has drivers for the following devices:

X **SwiftLink:** This driver uses the SwiftLink cartridge. The SwiftLink cartridge plugs into the cartridge port, and operates at up to 38,400 bps.

X **CommPort:** This driver uses the CommPort cartridge. The CommPort cartridge plugs into the cartridge port, and can operate at speeds up to 38,400 bps.

HART: This driver uses the HART cartridge. The HART cartridge plugs into the cartridge port, and can operate at speeds up to 57,600 bps.

UP9600: This driver, written by Daniel Dallman, uses a modified RS232 interface to achieve 9600 bps through the user port.

X If you select "I have no serial cartridge", Novaterm uses the User port driver:

User port: This driver uses the Commodore 64's built-in user port. It should be used if you have a modem directly connected to the user port (such as a 1650, 1660, or 1670), or if you are using a standard RS232 interface that plugs into the user port. The user port can only operate at speeds up to 2400 bps. Higher speeds are not allowed when using this driver.

1.1.3.1 Selecting a SwiftLink/CommPort address

If you specified a SwiftLink or CommPort as your serial device, you must specify its memory address. A SwiftLink is shipped with an address of \$DE00, but it may be moved by physically altering the cartridge, or by using an AddressFixer device (see Appendix C, **Vendor information**). A CommPort is shipped with an address of \$DF00, but it may be relocated in a similar manner. The following addresses are available:

\$DE00: This address should be selected for a standard SwiftLink, or for a CommPort mapped to \$DE00.

\$DF00: This address should be selected for a standard CommPort, or for a SwiftLink mapped to \$DF00.

\$DE20: This address should be selected for a SwiftLink or CommPort mapped to \$DE20.

\$DF20: This address should be selected for a SwiftLink or CommPort mapped to \$DF20.

\$D700: This address should be selected for a SwiftLink or CommPort mapped to \$D700. This may only be used on a C128 in 64 mode.

1.1.4 Selecting a dialing method

Select whether you wish the modem to dial with tone or pulse.

1.1.5 Selecting a C128 VDC screen driver (C128 in 64 mode only)

If you are running a C128 in 64 mode, and if you have an 80-column RGB monitor, you have the option of using the C128's 80-column screen. You can choose a display with 25, 28, 43, or 50 lines. If you choose 28-line or 50-line mode, you may need to enter terminal mode later and adjust your monitor to bring all the lines onto the screen.

If you can't use the C128's 80-column mode, just select "Give me the regular screen".

X If you are running a C64, this option will not be presented to you.

1.1.6 Specifying colors and fonts

If you did not or were not able to select a C128 80-column screen above, you are presented with the following options to configure Novaterm's bitmapped 80-column mode.

The next item allows you to set the default screen colors. Pressing RETURN repeatedly on any of the three options toggles through the 16 colors.

After the screen color menu, the configuration program asks you to choose between two types of 80-column printing styles. Choose whichever is easier to read. The "normal" font has characters three pixels wide, which may be hard to read but do have space between them. The "wide" font has characters four pixels wide, which may be easier to read but are crammed together because there is no space between each character.

The following item allows you to choose a color combination for 80-column mode from a predefined set of choices. Choose the one that is easiest for you to read.

1.1.7 Specifying device settings

If you have more than one disk drive, you are asked to specify which device should be used for the following operations (see section 4.7.2, **Device settings**):

- File transfers (uploading and downloading)
- Buffer file copy operations
- Printer and secondary address

1.1.8 Specifying the memory device

Finally, the configuration program asks you to specify a memory expansion device for the buffer, if you have one.

Novaterm has a device-independent interface for accessing memory, which means that Novaterm can use a variety of memory expansion devices for the buffer with the appropriate driver. Each driver uses all of the available memory in the device (except for the RAMLink/RAMDrive driver; see below). Each driver also has a maximum directory size (number of files) that may be stored in the buffer directory.

For information on where to obtain these devices, see Appendix C, **Vendor information** .

internal: Uses the internal memory of the Commodore 64. With this driver, your buffer is limited to 2.5K of internal memory. The maximum size of the directory is 22 files.

If you chose a C128 80-column screen previously (see 1.1.5), this option will instead select the **alt-internal** driver, which uses some of the memory not used by the C64 80-column emulation to give a total of 6.5K.

VDC+internal: Uses the internal memory of the Commodore 64 as well as the Commodore 128's VDC memory. This driver only works if you are using a Commodore 128 in 64 mode. It uses the 16K or 64K of RAM associated with the VDC chip (the driver detects how much is available), as well as the internal memory of the Commodore 64, for a total of 18.5K or 66.5K of space. The maximum size of the directory is 64 files.

If you chose a C128 80-column screen previously (see 1.1.5), this option will instead select the **VDC+alt-int** driver, which avoids using the first 16K of VDC memory (leaving it for the 80-column screen). If you only have 16K of VDC memory installed in your computer, this driver only uses the 6.5K of internal memory. If you have 64K of VDC memory, the driver uses the upper 48K of that memory, for a total of 54.5K of buffer space.

REU: Uses a RAM Expansion Unit (1700/1750/1764). The driver detects the size of the REU. (The driver will detect REU's expanded to 2 MB, and even more if it's ever done.) The maximum size of the directory is 144 files. The driver also allocates three pages (768 bytes) of memory for use with the **fast-REU 80-column driver** (see 4.2.1, **80-column driver**).

BBG-GEORam: Uses a BBGRam or GEORam. The driver detects the size of the unit (BBGRam cartridges are available in sizes up to 2

MB). The maximum size of the directory is 144 files.

RAMDrive: Uses a RAMDrive. This driver searches for a direct access (DACC) partition in your RAMDrive with the name "novaterm". The partition can be any number and any size; the driver detects its size and allocates the memory in the partition for the buffer. The maximum size of the directory is 144 files.

RAMLink: Uses a RAMLink. This driver searches for a direct access (DACC) partition in your RAMLink with the name "novaterm". The partition can be any number and any size; the driver detects its size and allocates the memory in the partition for the buffer. The maximum size of the directory is 144 files.

1.1.9 Dealing with a hardware conflict

Certain combinations of a serial device and memory expansion device result in a hardware conflict. This means that the two specified devices use the same addresses for access. This will cause problems when Novaterm or another program tries to access one; the other will be accessed at the same time.

To help avoid this situation, the configuration program notifies you when a selection of drivers would result in a conflict. It gives you the option of going back to choose a different driver for either the serial device or the memory device.

Section 4.8 provides more detail about hardware conflicts.

1.1.10 Selecting a real-time clock device

Novaterm provides several methods of setting the Commodore 64's internal clock from an outside source. Select from one of the following choices:

CMD-RTC: If you have a hard drive or floppy drive from Creative Micro Designs, Inc., or a RAMLink with a real-time clock installed, choose this method. This driver scans devices 8 through 30 for a CMD drive with a clock using the T-RB disk command. Novaterm reads the time from the first drive that does not report an error from this command.

BBRTC-port1: If you have a BBRTC device from Performance Peripherals, Inc. plugged into joystick port 1, choose this method.

BBRTC-port2: If you have a BBRTC device from Performance Peripherals, Inc. plugged into joystick port 2, choose this method.

Manual: If you do not have any of the above real-time clocks, choose `Manual`. This method asks you to enter the current time in 24-hour format.

All of the above drivers read and set the real-time clock when you start up Novaterm, except for the `Manual` option. If you have selected the `Manual` driver, you will not be prompted for the current time when you start up Novaterm; instead, you must select the driver again from Novaterm's Configuration menu. At that time, Novaterm prompts you for the current time.

1.2 Novaterm basics

1.2.1 Using menus

✓ All menus in Novaterm work the same way: Use the up/down cursor key to move a highlight bar up and down the options, and press RETURN to select a highlighted option.

You may also press a number that corresponds to an item, even though menu items are not labeled with numbers. Pressing the 1 key will instantly select the first item of a menu, pressing the 2 key will select the second item, and so forth. Pressing the numeric key has the effect of moving the bar to the corresponding item *and* pressing RETURN. If there are more than nine items on a list, then letters are used for the tenth item and beyond; the A key selects the tenth item, the B key selects the eleventh item, and so on.

X For extra convenience, the F1 key always selects the last item of a menu. Usually, the last item is Quit or Main menu, which leaves the menu. Therefore, you may always use the F1 key to jump out of the current menu. Even if a given menu does not have Quit or some variant as the last item, you may still use the F1 key to leave the menu.

1.2.2 Entering text ✓

From time to time, Novaterm will ask you to enter text in response to some command. (For instance, you will often be asked to enter a file name.) When entering text, you may use the DEL key to back up and correct any mistakes. Pressing RETURN finishes your entry.

Sometimes, you will see some text already in the area where you want to type. This usually happens when you are asked to change a parameter, and the previous value of that parameter is displayed for you. In this situation, the right-arrow key will function just as you'd expect; it will keep the character that appears under the cursor and

make it part of your new entry.

1.2.3 The modular approach

Modules are single files all of the same type. They are used to support different hardware, or different features that share the same basic functions. For instance, modems all do the same basic things: they can dial a phone number, send and receive data, detect a carrier signal, and so forth. However, different modems may require different programming to perform the same tasks. With Novaterm, a module may exist for each type of modem. Each function (such as dialing) has a common access point, so the rest of the software doesn't care what type of modem you specify.

The other modules behave the same way. Below is a summary of all of the different types of modules used by Novaterm. You'll learn about them in other sections of the manual.

<u>Prefix:</u>	<u>Description:</u>	<u>Section:</u>
80col	80-column drivers	4.2.1
asc	ASCII translation tables	7.2
font	40-column fonts	4.2.2
font80	80-column fonts	7.3
modem	Modem types	4.4.1
nova	Utility programs	7
prt	File transfer protocols	3.4.3
ram	Memory expansion drivers	1.1.7
serial	RS232 serial drivers	1.1.3
term	Terminal emulations	3.1.1
time	Time setting modules	1.1.10
tmod	Scripts to load emulations	3.1.1

2. GETTING FAMILIAR

If you've never used Novaterm before, take some time to get familiar with the software. This section takes you on a quick tour of Novaterm. More specific information on each part of the software may be found in later sections.

2.1 The Main Menu: a quick tour

Whenever you load up Novaterm, you always start at the Main Menu. This menu is the "launch point" for accessing the various functions of the system. In the course of using Novaterm, you'll often return to the Main Menu in order to get to these functions.

To get a feel for Novaterm, it is helpful to understand what happens when you select each item on the Main Menu.

2.1.1 Terminal mode

Move the menu bar up to `Terminal mode`, the first item on the Main Menu, and press `RETURN`.

The screen clears, and a single line of information appears across the top of the screen. A cursor appears on the second line. This is terminal mode.

Terminal mode is where you interact with a computer on the other end of your modem. The entire screen is dedicated to displaying text. In terminal mode, anything you type on the keyboard is sent to your modem, and the data received by your modem from the other computer is displayed as it arrives.

If you have a Hayes compatible modem, you may be able to see how this works. Watch the screen while you type a few keys. Your modem should echo the keys you type. Novaterm receives these characters from the modem and displays them on the screen. If your modem is not Hayes compatible, you'll have to wait until you are on-line with another computer for a demonstration of terminal mode.

To exit terminal mode and return to the Main Menu, press `C= Z` (hold down the `C=` logo key and press the `'Z'` key at the same time).

2.1.2 Dial a number

Move the menu bar to `Dial a number` and press `RETURN`.

At the top of the screen, a list of commands appears. At the bottom of

the screen is the label `Phone book:` with the words `nova phone` after it.

This is the dialing directory. On this menu, you may create entries for bulletin boards you call frequently. Each entry may have a name, phone number, password, and other information associated with it. Your entries will show up in a menu below the commands, as soon as you create some.

Press `F2` to take a look at the menu for editing a single entry. (Select `Quit` to return to the dialing directory.)

Press `F1` to return to the Main Menu.

2.1.3 Configuration

Move the menu bar to `Configuration` and press `RETURN`.

This menu contains all of Novaterm's configuration options. You set some of these options when Novaterm led you through the first-time configuration process. All of these options are covered in a later section of the manual.

Press the cursor-right key. A second set of configuration options appears. There are two parts to the Configuration menu; the cursor-right key toggles between them.

Select `Quit` to return to the Main Menu.

2.1.4 Disk operations

Move the menu bar to `Disk operations` and press `RETURN`.

This menu provides some convenience functions for working with the disk drive. From here you may enter disk commands, look at the disk directory, and view the contents of text files.

Move the menu bar to `Disk directory` and press `RETURN`. The directory of the disk is listed on the screen.

Select `Quit` to return to the Main Menu.

2.1.5 Buffer menu

Move the menu bar to `Buffer menu` and press `RETURN`.

The 'buffer' is a section of computer memory reserved for storing data received by the modem. The buffer is most useful for saving incoming text that you are interested in looking at later. Novaterm

won't always store data in the buffer; you control when the "buffer capture", as it's called, is active.

Remember the "RAM device driver" modules (section 1.3.3)? Novaterm may use a memory expansion device as storage for the buffer. If you have a Commodore RAM Expansion Unit (1700/1750/1764), a RAMLink or RAMDrive, or a BBG/GEORam cartridge, you'll have a lot of room available for capturing text.

The buffer menu provides options for working with the contents of the buffer. Novaterm's buffer actually behaves like a disk. It has a directory, and files are stored with individual file names. You can create, rename, and scratch files, as on a normal disk.

Select `Quit` to return to the Main Menu.

2.1.6 Device settings

Move the menu bar to `Device settings` and press `RETURN`.

This menu specifies which disk device to use for each of the listed operations. For instance, if you have a drive 8 and a drive 9, you could specify that modules and configuration files be loaded from drive 8 (the "program" drive) and file transfers use drive 9 (the "upload" drive and "download" drive).

Select `Quit` to return to the Main Menu.

2.1.7 Utility modules

Move the menu bar to `Utility modules` and press `RETURN`.

This menu lists a set of separate programs that expand on Novaterm's basic functionality. When you select one, Novaterm loads and runs the program for you.

Move the menu bar to `Text editor` and press `RETURN`. After a few moments, Novaterm's text editor appears on your screen. This is a general-purpose text editor for creating sequential files. Press `Ctrl-H` to see a summary of commands that work in the text editor. Press `Ctrl-Z`, then answer `yes` to leave the text editor. On the menu that follows, select `Return to Novaterm` to return to Novaterm.

2.1.8 Exit terminal

Move the menu bar to `Exit terminal` and press `RETURN`. This option will terminate the Novaterm session. Answer `no` to return to the Main Menu.

3. ON-LINE ACTIVITIES

This section takes you through the functions most commonly used while on-line. It starts with a look at terminal mode, the center of your on-line activities. It then explains how to set up the dialing directory, how to use the buffer to capture text, and how to transfer files (upload and download).

3.1 Exploring terminal mode

Terminal mode is where you interact with a computer on the other end of your modem. The entire screen is dedicated to displaying text. In terminal mode, anything you type on the keyboard is sent to your modem, and the data received by your modem from the other computer is displayed as it arrives.

In terminal mode, Novaterm can display text in 40- or 80-column mode. 40-column mode is the normal state for a Commodore 64. Novaterm also provides an 80-column mode, where each character is half the width of a 40-column character. See section 3.1.6, **Notes on 80-column mode** for more information about it.

This section covers various features of terminal mode, including the status line, key commands, key reassignments, and terminal emulations.

3.1.1 Terminal emulations

Terminal mode depends on the use of "terminal emulations" to display incoming text. A terminal emulation is a special driver (module) that interprets incoming text and displays it appropriately.

For instance, one terminal emulation shipped with Novaterm is called "ANSI". ANSI is the name of a commonly used emulation where certain sequences of characters are used to position the cursor, change text color, clear the screen, and do other display-related things. In ANSI, whenever the driver receives the ESCape character, the characters immediately following will always be some sort of special ANSI command. Novaterm's ANSI driver knows how to interpret these commands and take the appropriate action.

So why are they called "emulations"? In the old days, almost all computers were big mainframes, and they had one or more dumb terminals attached to them. These terminals interpreted certain escape sequences in the manner described above. Today, these escape sequence command have been adopted by the on-line community.

When a desktop computer interprets sequences in this way, it is said to “emulate” the old dumb terminals.

Typically, for each on-line service you call, you'll need to determine which terminal emulation to use. Novaterm includes a variety of terminal emulations:

ANSI: This emulation displays ANSI color and graphics in 80-column mode. This should almost always be used when calling IBM-based bulletin boards.

ANSI-40: This emulation displays ANSI color and graphics in 40-column mode. This type of display is limited, as ANSI graphics are typically laid out for 80-column displays, and will not look correct in 40 columns. ANSI-40 is intended for users who want the benefits of ANSI colors (and some graphics) without using 80-column mode.

VT102: This emulation supports the VT100 and VT102 emulations. It is typically used when dialing into large computer systems, such as an Internet provider or a public library's on-line catalog.

VT52: This emulation is typically used when calling large computer systems that do not support VT100 or VT102. (It is an older standard.)

Standard-80: The Standard and Commodore emulations are simple text displays; they do not actually interpret any special escape sequences. Every character received by the modem is displayed as is (with ASCII translation, of course). This is appropriate for any on-line service that does not support terminal emulation at all.

Standard-80 is, of course, the 80-column version.

Standard-40: This is the 40-column version of Standard.

Commodore: This emulation is for using bulletin boards that can display Commodore color/graphics. The only difference between this and Standard-40 is that ASCII translation is turned off.

ANSI-wide, VT102-wide, Std-80-wide: These are all variations of the above emulations that use the wide 80-column font.

3.1.2 Status line

In terminal mode, Novaterm displays a status line which reports various conditions. In 40 columns, the status line looks like this:

```
:E:T:K:C:B:X: 2400 00:00:00
```

In 80-column mode, the status line has some extra information:

```
:E:T:K:C:B:X: 2400 ANSI Zmodem 3:30pm 00:00
```

The letters separated by colons represent the status of certain terminal settings. A letter appears highlighted (reversed) if its setting is "on" and normal if its setting is "off". The function of each letter is described in more detail in the referenced sections:

E: Local echo. This determines whether keys you type are displayed on your screen as well as sent to the modem. (see 3.1.3, **Commodore key commands** , C= E)

T: ASCII translation. This indicates whether ASCII translation is in effect. (see 3.1.3, **Commodore key commands** , C= A)

K: Control character display. If this is on, control characters are printed explicitly. For instance, a CTRL-A would be displayed with the characters ^A. (see 3.1.3, **Commodore key commands** , C= K)

C: Carrier indicator. This indicates whether the modem is on-line.

B: Buffer capture. This indicates whether data is being captured to the buffer. (see 3.1.3, **Commodore key commands** , C= B, C= O)

X: Script file active. This indicates whether a script file is currently running. (see section 6, **Scripts**)

The number directly to the right of the status letters displays the current baud rate.

The clock in the right-hand corner in both 40- and 80-column modes is an on-line timer. It is reset to zero and begins counting immediately after a modem connection is established. When the modem is disconnected, the timer stops and retains its last value until a connection is made again. The timer is useful when you are calling long distance or are using an on-line service that charges by the minute or hour. In 40-column mode, the clock counts seconds; in 80-column mode, the clock updates every minute.

In 80-column mode, the extra width available on the status line is used to display the current terminal emulation (see 3.1.1, **Terminal emulations**), file transfer protocol (see 3.4.3, **Selecting a protocol**), and the current time (see 1.1.10, **Selecting a real-time clock device**).

The status line may be turned off by pressing C= S in terminal mode (see section 3.1.3, **Commodore key commands** , C= S).

3.1.3 Commodore key commands

While using terminal mode, a variety of functions are available to you by pressing logo key combinations. The terminal mode key commands are summarized below. A key described as C= X means you should hold down the Commodore logo key while pressing the letter X.

NOTE: Many of the descriptions below refer to features covered in later sections of the manual. If you read something that doesn't make sense to you, follow the reference to learn more about it.

C= A - Toggles ASCII translation. This command determines whether incoming data should be translated from ASCII to PETSCII for proper display on the C64. This should always be on unless you are connected to a Commodore color/graphics BBS. This command works only with the "Standard" terminal emulations, as the others require translation to remain on to function properly (see 3.1.1, **Terminal emulations**).

C= B - Pause/resume buffer capture. This is only valid when a buffer capture file has been opened using C= O. Pressing C= B stops text capture while leaving the capture file open. The highlighted B in the status line turns off, indicating that text is no longer being captured. Pressing C= B again resumes text capture and highlights the B.

C= C - Toggle between 40 and 80 columns (see 3.1.5, **Notes on 80-column mode**). This command works only with the Standard terminal emulations, as the others require the 80-column display to function (see 3.1.1, **Terminal emulations**).

C= D - Download a file. (see 3.4.5, **Receiving a file**)

C= E - Toggle local echo. When local echo is on, everything you type is sent to the screen as well as to the modem. This setting is indicated by the letter E in the status line. Local echo should be turned on if you are connected to a computer or bulletin board which does not echo back characters. Most bulletin boards echo characters, but if you are connected directly to another terminal (if, say, you and a friend were linked up to exchange files) and you can't see what you are typing, you should turn local echo on.

C= F - Display the amount of free disk space.

C= G - Execute a script file. (See section 6, **Scripts**.)

- C= H - Hang up phone. This hangs up the modem.
- C= I - Hayes modem initialization. This command sends the modem initialization string to the modem only if you have specified a Hayes-compatible modem in your configuration.
- C= J - Pick up phone. This takes the modem off the hook.
- C= K - Toggle control character display. When this is turned on, any control characters printed in terminal mode are displayed explicitly. Control characters are those characters generated by holding down the CTRL key while pressing another key. Normally, you cannot see them when the terminal tries to print them, because they are reserved for performing display functions. With this option on, however, Novaterm displays the CTRL-A character with the two symbols ^A, CTRL-B with ^B, and so forth. The only exceptions are carriage return and backspace, which retain their normal functions. When control character display is turned off, any control characters that are received have their normal effects. The K in the status line indicates whether this is on or off.
- C= L - Load a new function key file (see 3.1.4, **Function keys**).
- C= M - Display a menu of these commands on the screen.
- C= N - Display the amount of free buffer space in bytes and equivalent disk blocks.
- C= O - Open/close buffer capture file. Pressing C= O prompts you for a file name, and then opens a file in the buffer for receiving text. Any text that is subsequently displayed on the screen is stored in the buffer file as well. Pressing C= O again closes the buffer file.
- C= P - Select protocol. (see 3.4.3, **Selecting a protocol**)
- C= Q - Enter chat mode. This option works in the Standard terminal emulations only (see 3.1.1, **Terminal emulations**). In a Standard terminal emulation, this command provides a 159-character keyboard buffer (at the top of the screen in 40 columns; at the bottom in 80 columns) to use in conference chat modes.
- C= R - Set baud rate. (see 4.1.2, **Baud rate**)
- C= S - Toggle status line. In 40 columns, turning the status line off frees up an extra line at the top of the screen for text display.
- C= T - Select terminal. (see 3.1.1, **Terminal emulations**)

C= U - Upload a file. (see 3.4.6, **Sending a file**)

C= V - View disk directory.

C= W - Toggle scroll-ahead. In 80-column mode, scrolling is normally done by scanning the incoming data stream for carriage returns, and scrolling the screen several lines at a time in anticipation. This is done to make the scroll more efficient. This option allows you to turn the feature off, for situations where the multiple-line scrolling interferes with how text is supposed to be arranged on the screen.

C= X - Send break signal. This produces a 250 millisecond break in the data signal. On many on-line service providers, a break signal causes an interruption of some sort.

C= Y - Restore default attributes. This changes the screen and cursor back to the colors previously set in the Configuration menu (see 4.2, **40/80-column settings**). In 80 columns, this command also turns off reverse and underline modes. If you received some funny characters that changed your text color, just press C= Y to return to normal.

C= Z - Exit terminal mode, or abort script. Normally, this command leaves terminal mode and returns you to the Main Menu. If a script is running, however, this command stops the script, but keeps you in terminal mode. You must then press it again to return to the Main Menu.

C= + - List menu on printer. This command lists a menu of these commands on your printer.

C= @ - Disk command. Prompts for and executes a disk command from terminal mode.

C= - - Toggle transfer translation. (see 3.4.8, **Transferring text files from non-Commodore systems**)

C= 0 - Enable/disable Commodore keys. When you press C= 0 (zero), a caret symbol appears in the status line. This indicates that the key commands are disabled and become normal graphics characters again. This allows you to type graphics characters, say, for creating a graphics screen in a message on a BBS. Pressing C= 0 again removes the caret symbol and changes the keys back to commands.

3.1.4 Function keys

The F1 through F8 keys on the Commodore 64 keyboard can be

redefined to send strings to the modem when you press them in terminal mode. These are useful for defining strings that you find yourself typing often (such as your user name when logging into an on-line service). Novaterm calls them "function keys" or "macro keys".

There are 16 function keys available. F1 through F8 are accessed as shown on the keyboard. The remaining function keys are accessed by pressing either CTRL or the Commodore logo key in combination with the function keys, in the following order:

F9: CTRL-F1	F13: CTRL-F5
F10: C= F1	F14: C= F5
F11: CTRL-F3	F15: CTRL-F7
F12: C= F3	F16: C= F7

The F1 key is reserved for BBS password definitions, and does not appear on this menu (see 3.2.1, **Creating a dialing entry**).

The function keys are defined from the F-key definition menu. (See 4.7.1, **Define F-keys** for more information.)

3.1.5 Key assignments

There are a few characters that do not exist on the C64 keyboard. These are characters that are part of the standard character set used by the rest of the world. To accommodate these characters, some of the graphics keys have been redefined.

<u>To get:</u>	<u>Character:</u>	<u>Type:</u>
escape key		left arrow
tab key		RUN/STOP or CTRL-I
backspace (8)		INST/DEL
true delete (127)		CLR/HOME
backslash	\	British pound
underline	_	shift-@ (at)
left brace	{	shift+ (plus)
right brace	}	shift-minus
vertical bar/pipe		shift-British pound
back apostrophe (grave)	`	shift-* (asterisk)
tilde	~	shift-↑ (caret/up arrow)

In addition to the above characters, the entire lowercase European character set is available. In order to use these characters, you must be using either ANSI or VT102 emulation, have local echo turned off, and have the Commodore command keys disabled by pressing C= 0

(zero) (see 3.1.3, **Commodore key commands** , C= 0).

European character set and key mappings

Key	char	Key	char	Key	char	Key	char
C= Q	è	C= A	à	C= X	Ð	C= *	Û
C= W	é	C= S	á	C= C	ç	C= 1	ı
C= E	ê	C= D	â	C= V	Æ	C= 2	ı̇
C= R	ë	C= F	ä	C= B	ß	C= 3	ø (Ö)
C= T	ý	C= G	å	C= N	ñ	C= 4	ò
C= Y	ÿ	C= H	ã (Ä)	C= M	ü	C= 5	ó
C= U	ì	C= J	ù	C= +	æ	C= 6	ô
C= I	í	C= K	ú	C= -	Ë	C= 7	ö
C= O	î	C= L	û	C= £	£	C= 8	ø
C= P	ï	C= Z	ö (Ü)	C= @	Ö		

3.1.6 Notes on 80-column mode

If you run Novaterm on a C128 in 64 mode, and if you have an 80-column RGB monitor, you can use the C128's 80-column screen for terminal mode. When you switch to terminal mode with one of the C128 VDC drivers in place, the 40-column screen goes blank. At this point, you must switch your monitor to 80-column RGB mode. When you leave terminal mode, the main menu reappears on the 40-column screen, so you must switch your monitor back to 40-column mode.

The C128 VDC screen is a true 80-column display with easily readable text and a fast display. The notes below do not apply to this mode.

On a C64, Novaterm simulates an 80-column display by drawing half-width text characters on the C64's high-resolution bitmap screen. Because the C64 was not designed for 80 columns, there are a few limitations to be aware of. Novaterm compensates for most of these limitations if you choose the proper settings.

3.1.6.1 Improving readability

First of all, the text can be just plain hard to read. Each character is only four pixels wide, and vertical lines are one pixel in width. Your monitor must be in decent shape to read it. If your monitor is blurry at all -- or if you are trying to use a TV set as a monitor -- you'll have a difficult time. Make sure your monitor is adjusted properly (by fiddling with the control knobs).

If you choose to replace your monitor, Commodore-brand color monitors do an excellent job of displaying 80-column mode. Monochrome monitors -- like the old "green screens" -- are even sharper, but you will lose the color.

You may also try changing the text and background color in 80-column mode (see 4.2.4, **80-column color**). The default is light-gray on black. Your eyes may have an easier time with a different combination. A black or dark-gray background is needed for proper color reproduction with ANSI terminal emulation, but if you primarily use 80-column mode for VT102, VT52, or standard emulation, a black-on-yellow or black-on-white display may be preferable. Some people prefer light-green text on a medium-gray background. Another good combination is black text on medium- or light-gray background.

Turning up the brightness knob on your monitor helps with any color combination. It lessens the "color bleed" problem of single-pixel characters, where dark colors tend to bleed over into light ones.

One more thing you can do -- and you might think this is really boneheaded, but it works -- is keep your screen clear of dust. You'll be surprised what a difference a clean monitor makes.

3.1.6.2 Compensating for slow display speed

Because Novaterm must draw each character in the bitmap, the printing speed of 80-column mode is noticeably slower than 40-column mode. Novaterm's 80-column mode can keep up with incoming data at 4800 bps; at 9600 bps, it falls behind and starts to lose text *unless* you use one of the options described below.

If you are using Novaterm at speeds in excess of 4800 bps, chances are you are using a high-speed modem (9600, 14.4k, or 28.8kbps). All high-speed modems support a feature called "hardware flow control". This feature is explained in more detail in section 4.6.2, **Flow control**. Using hardware flow control allows the 80-column display to keep up with incoming text at all times. You should have hardware flow control turned on in Novaterm (and in your modem). With hardware flow control, you can set Novaterm's baud rate as high as 38,400 bps without losing any data.

Another option is available if you use a RAM Expansion Unit (1700/1750/1764). An 80-column driver is provided that implements a fast scroll using the REU's direct memory access (DMA) capability. The first-time configuration program automatically loads this for you

if you specified REU as your choice of memory expansion. See section 4.2.1, **80-column driver** for information about using the fast-scrolling 80-column driver.

3.1.6.3 Displaying color

One limitation of the Commodore 64 is that it cannot display more than two colors in a given 8x8 pixel area. (It can in multicolor mode, but this mode cuts the horizontal resolution in half, so it can't be used for 80 columns.) Therefore, every two adjacent characters in 80-column mode must be the same color. This can cause some characters to appear in a different color than intended by the on-line service. This explains why some ANSI color/graphics displays will appear distorted when displayed in Novaterm's 80-column mode.

3.2 Dialing directory

Novaterm provides a phone directory for keeping track of on-line services you call frequently. In the phone directory, you may create an entry for each number you call. You have the opportunity to define several configuration options for each entry. When you dial an entry and make a connection, the configuration options you specified are automatically set for you.

You may enter a maximum of 19 entries. However, you may keep separate lists of entries stored in different files called "phone books". Phone books are useful if you have more than 19 entries to store, or if you want to keep some entries logically separated. For instance, when I moved back and forth between Seattle and Los Angeles for school, I kept my list of Seattle numbers in a file called `seattle phone` and my list of Los Angeles numbers in `la phone`.

3.2.1 Creating a dialing entry

To get started with a new phone book, press F2 to edit the first blank entry. Novaterm brings up the entry editing menu. Most of the fields will be blank for a new entry. Move the menu bar down to each item and enter values.

Each item has the following meaning:

Name of entry : Name of the BBS or on-line service.

Phone number: Phone number to be dialed. When entering a phone number, a comma imbedded between two digits represents a one-second delay.

Password: The string entered here is automatically defined in the F1 function key when you dial and connect to this number.

Term emulation: This is the terminal emulation to be used with the on-line service.

Baud rate: Novaterm switches to this baud rate just before dialing the number.

Word/parity: Novaterm changes the word length and parity to the setting specified here when a connection is established. This can be one of three settings: 8 bits/no parity/one stop bit (8N1), 7 bits/even parity/one stop bit (7E1), or 7 bits/odd parity/one stop bit (7O1).

Connect script: This item is optional. It specifies the name of a script file to execute when a connection is established. This is most useful for specifying auto-login scripts (see section 6, **Scripts**).

Select **Quit** when you are finished. Since you made changes to this entry, Novaterm asks you if you'd like to save them to disk. If you do, answer **yes**.

You may create more entries by moving the highlight to the blank line below the last entry and pressing F2.

3.2.2 Directory commands

If you have created some entries, you should now see your list on the screen. Use the up/down-cursor key to move the highlight to one of the entries. You may use the function keys to access the following commands, which act on the highlighted entry:

F1: Return to the main menu.

F3: Insert and edit a blank entry before the highlighted one.

F4: Delete the highlighted entry.

F5: Edit the configuration for the entry.

F6: Type in a number to be dialed.

F7: Save the current list to a phone book.

F8: Load a different phone book.

3.2.3 Dialing a number

To dial one of your entries, move the highlight down to the entry and press **RETURN**. This starts the dialing process.

Before dialing the number, Novaterm first loads the terminal emulation for the entry. It also loads the Connect Script if one was

specified.

Novaterm then starts to dial the number. The screen displays the number being dialed, along with a timeout countdown. If a connection cannot be made within the timeout period, Novaterm gives up and dials again (see 4.3.4, **Redial time**).

If you have a Hayes compatible modem, Novaterm will detect any response codes reported by the modem while waiting for a connection. Most Hayes compatible modems will respond with at least some of the following messages:

NO CARRIER - The modem timed out and gave up.
NO ANSWER - Same as NO CARRIER.
NO DIALTONE - The modem did not detect a dial tone.
BUSY - The number dialed was busy.
VOICE - Someone picked up the line on the other end.

If Novaterm detects any of these response strings from the modem, it immediately stops the current attempt and dials again. If your modem can detect a busy signal, you can redial a busy on-line service quickly, increasing your chances of grabbing their line when it is free.

While waiting for a connection, you may press one of the following keys to interrupt the process:

Space - Give up this attempt and dial again.
F1 - Quit dialing and return to terminal mode.

When the modem finally makes a connection, Novaterm changes your configuration settings, executes the Connect Script if there is one, and jumps into terminal mode. At this point, you are ready to start communicating.

3.2.4 Multiple dialing

You have the option of dialing more than one entry in succession.

On the dialing directory, use the space bar to tag several entries for multiple dialing. You'll see the > symbol appear next to tagged entries. When you press RETURN, Novaterm will start dialing with the first entry.

If your modem encounters a busy signal or does not establish a connection during the timeout period, Novaterm gives up and starts dialing the next number in the list that you selected. Novaterm continues to loop through the list until a connection is established with

one of the selected entries.

3.3 Using the buffer

The “buffer” is a section of memory used by Novaterm for the purpose of storing incoming text. When set up to do so, Novaterm “captures” text in the buffer at the same time it displays the text on the screen in terminal mode. The text is stored in a “capture file”.

Novaterm’s buffer can do more than simply capturing text. File transfers to the buffer are possible as well (see 3.4, **File transfers**). In fact, Novaterm’s buffer is set up as if it were a disk drive. It has a directory and keeps track of distinct files. The only difference is that its data is kept in memory. (This kind of storage is called a “RAMdisk”.)

Since the buffer isn’t really a disk drive, the Buffer Menu provides disk-like operations. Most of the standard operations that work with a disk are available, such as scratching, renaming, and copying files.

This section covers the operation of the buffer and explains the various options available on the Buffer Menu.

3.3.1 Specifying the memory device

On the Buffer Menu, move up to the `Memory device` option and press `RETURN`. Novaterm brings up a list of memory device drivers. You should use the one corresponding to the hardware you have connected. See section 1.1.7, **Specifying the memory device** for a description of each driver.

3.3.2 The buffer directory

Select `Buffer directory` from the Buffer Menu to list the buffer directory. The directory is listed just like a disk directory, with block sizes that indicate how many blocks the file will use when copied to a physical disk. The directory also lists the exact byte size of each file, and the number of bytes and equivalent blocks free.

Files in the buffer directory can have type `prg`, `seq`, or `scr`. `usr` files are listed in the buffer directory with type `scr`, to indicate a script file.

3.3.3 Working with buffer subdirectories

Because the buffer can potentially have a large number of files, Novaterm uses a simple subdirectory scheme for organizing files.

Each file can reside in a particular subdirectory, numbered 0 through 30. You use the `Change subdirectory` option to specify the current subdirectory. The `Buffer directory` option will list only those files in the current subdirectory.

For instance, if you have a number of script files loaded in the buffer, they could clutter up the directory, and a capture file you created would show up along with the scripts. Therefore, by default, Novaterm loads and executes scripts from subdirectory #1, while buffer captures use the current subdirectory (which is set to #0 when you start up Novaterm).

Furthermore, by changing the Device Settings for the upload and download device, you can direct file transfers to use yet a different buffer subdirectory, (see 4.7.2, **Device settings**).

3.3.4 Capturing text in the buffer

Text captures are initiated from terminal mode. To capture text, switch to terminal mode, and follow these steps:

1. Press `C= O` (see 3.1.3, **Commodore key commands**, `C= O`).
2. Type in the file name of the capture file. This file is opened in the buffer. From this point on, any text that is displayed on the screen is also stored in the buffer capture file.
3. If you want to keep the capture file open but temporarily stop data from being stored in it, press `C= B`. The `B` in the status line turns off, indicating the capture is paused. Press `C= B` again to resume the capture.
4. When you have captured all the text you want, press `C= O` again to close the capture file.

The capture file now exists in the buffer directory. You can switch back to the Buffer Menu and use any of the options described below on the file. (Of particular note after the text capture is the `Save files to disk` option.)

3.3.5 Downloading a file to the buffer

To download a file to the buffer (using a file transfer protocol), you must change the `Download device` on the `Device settings` menu to the buffer. See section 4.7.2, **Device settings** for information on how to do this.

Once you have done this, follow the directions outlined in section 3.4, **File transfers** to download files.

3.3.6 Buffer file options

The Buffer Menu provides a number of disk-like operations for working with buffer files, which are described below.

3.3.6.1 Buffer directory

This option lists the buffer directory.

3.3.6.2 Disk directory

This option lists the directory of the current disk device.

3.3.6.3 Change buffer device

This is a convenience option that lets you change the buffer device. This change is also reflected on the Device settings menu (see 4.7.2, **Device settings**).

3.3.6.4 Load files from disk

This option brings up a list of files on the current disk device. Select one or more files to be copied into the buffer directory.

3.3.6.5 Save files to disk

This option brings up the list of files in the buffer. Select one or more files to be copied to the current disk device.

3.3.6.6 Rename file

This option lets you rename a file in the buffer directory.

3.3.6.7 Scratch file

This option deletes a file from the buffer directory.

3.3.6.8 Clear subdirectory

This option deletes all files from the current buffer subdirectory.

3.3.7 Buffer file processing options

The Buffer Menu also contains a series of options for special processing of files in the buffer.

3.3.7.1 View file

This option displays the contents of a buffer file.

3.3.7.2 View in 80 columns

This option displays the contents of a buffer file using 80-column mode.

3.3.7.3 Print file

This option prints the contents of a buffer file to the printer.

3.3.7.4 File to modem

This option sends the contents of a buffer file to the modem, line by line. After each line is sent, Novaterm pauses for the number of seconds specified in the `Line pacing` option on the Configuration menu (see 4.6.6, **Line pacing**).

3.3.7.5 ASCII to PET

This option translates a buffer file from standard ASCII to Commodore PETSCII. If you downloaded a file and forgot to use the "Transfer translation" option when you should have, this command will convert the file for you (see 3.4.8, **Transferring textfiles from non-Commodore systems**).

3.3.7.6 PET to ASCII

This option translates a buffer file from Commodore PETSCII to standard ASCII, placing carriage return/linefeed combinations at the end of each line. (This format is appropriate for MS-DOS text files.)

3.3.7.7 PET to Unix

This option translates a buffer file from Commodore PETSCII to standard ASCII, placing only linefeeds at the end of each line. (This format is appropriate for Unix text files.)

3.3.7.8 UUdecode file

A "uencoded" file is a file that has been converted from its regular form to an encoded form that contains only printable ASCII characters. This allows the file to be exchanged through a medium that only allows printable ASCII characters, such as a public message base or an e-mail message. An end user can "udecode" the file to obtain the original file. It is most commonly used on the Internet to include files in Usenet or e-mail messages.

This option will correctly decode uencoded files that were either

downloaded -- with the encoding still in ASCII -- or captured -- with the encoding converted to PETSCII.

When the original file name is detected, you are asked to specify its file type. If you select `seq`, you are asked whether to ASCII-translate the original file during the decoding process. Answer `yes` if you know that the original file is a standard ASCII text file. If you're not sure, just answer `no`; you can always go back and translate the original file with the `ASCII to PET` option (see 3.3.7.5, `ASCII to PET`).

3.3.7.9 UUencode file

This option uuencodes an original file. The encoding is stored in standard ASCII. This option is appropriate if you intend to upload the file directly.

3.3.7.10 UUencode in PET

This option uuencodes an original file, converting the encoding to Commodore PETSCII. This option is appropriate if you intend to put the file into a message editor using the `File to modem` option (see 3.3.7.4, `File to modem`).

3.3.8 Editing a file in the buffer

You can edit a file in the buffer using Novaterm's Text Editor program. If you launch the Text Editor from Novaterm's Utility Modules menu, the editor will have access to the buffer, and you can load and save files directly to it. See Appendix A, `Text Editor` for information on how to do this.

3.3.9 Buffer recovery feature

All of Novaterm's buffer drivers have a recovery feature built into them. If you load up a Utility Module without saving buffer files, or if you accidentally press your computer's reset button, Novaterm will retain the information in the buffer and make it available when you start it up again. Basically, as long as you do not turn off the computer completely, or use another program that overwrites portions of whatever memory you use for the buffer, it will remain intact if you quit Novaterm and return to it later.

If you are using some form of battery-backed or externally-powered memory expansion device, such as the BBGRam, then you can even turn the computer off. When you turn it on again and load Novaterm, the files you had stored in the buffer will still be there.

3.4 File transfers

Transferring files between computers is one of the oldest telecommunication activities. When transferring a file over a modem, the main concern is that the file gets transferred intact, unaffected by any noise on the telephone line that might corrupt the file in transit. To this end, every transfer uses some sort of file transfer “protocol”, which is just an agreed-upon method of conversation between two computers to ensure that every portion of a file is transferred correctly. All file transfer protocols use some sort of checksum to make sure data is being transferred without corruption, and some protocols can handle several files at once, making sure the file names remain the same on the computer receiving the files.

To clear any confusion: To “upload” a file means to *send* it; to “download” a file means to *receive* it.

Novaterm allows you to send and receive files from another computer using any one of a variety of popular file transfer protocols. This section covers the steps you must take to accomplish this.

3.4.1 Getting a file from an on-line service

If you are connected to an on-line service, you must somehow request a file to download, or request to upload a file. How you do this depends completely on the service itself. You'll have to check the bulletin board's help menus to find out.

3.4.2 Transferring files with a friend

If you have a friend who also has a computer and modem, you may transfer files with him or her by establishing a modem connection and following the steps below. Your friend doesn't have to use Novaterm or even a Commodore 64; any modem and communications software will do, provided you both choose the same file transfer protocol (see below).

3.4.3 Selecting a protocol

Before transferring a file, you must select a file transfer protocol. The on-line service (or your friend) on the other end of your modem must be set up to use the *same* protocol. If you're using an on-line service, check the help menus to find out how to do this. A file transfer will only when both ends of the link are using the same protocol.

To select a protocol in Novaterm, use the `C= P` command from Terminal Mode (see 3.1.3, **Commodore key commands**, `C= P`).

This command will bring up a list of Novaterm's protocol drivers. Selecting one from the list will load the protocol driver into memory. Novaterm will then use that protocol for file transfers.

How do you choose which protocol to use? Some protocols are better than others; you want to use the best one possible. However, you are constrained by the fact that the computer on the other end of the line must be using the same one. Therefore, you must choose a protocol that the other computer can also use.

The protocols supported by Novaterm are listed below in order of performance and convenience.

3.4.3.1 Zmodem

Zmodem is by far the most popular and superior file transfer protocol used in the on-line world today. Zmodem supports excellent error checking and batch file transfers (several files at a time). File names are transferred from the sender to the receiver, so the receiver doesn't have to ask the user to specify file names. Zmodem transfers in data blocks of up to 1024 bytes. Zmodem doesn't suffer from the problem of "file padding" like other protocols (as explained in the section on Ymodem).

Zmodem also has a nice feature called **auto-download**, where Novaterm automatically starts a download when it detects that the other computer has initiated an upload.

If you were a seasoned user of Novaterm 9.5, you'll remember that the Zmodem protocol included with that version could receive files, but not send them. The Zmodem protocol in Novaterm 9.6 can both send and receive files.

Novaterm's Zmodem also supports **crash recovery**, which allows you to resume an interrupted file transfer. If a Zmodem transfer was aborted early for any reason, you may continue the transfer where it left off by simply attempting to download the file again. If Novaterm detects that the file already on the disk - and if the `Auto-replace downloads` option is turned off (see 4.4.6) - it notifies the sender to continue where the file left off. (The only time a crash recovery isn't possible is when the file was downloaded to the buffer, *and* other buffer files have been created since the download. The aborted file must be the last file listed in the buffer directory for a crash recovery to be possible.)

Novaterm's Zmodem will operate in "streaming" mode *if* you are

downloading a file directly to the buffer. Zmodem defaults to streaming mode under the following conditions:

- The Allow streaming mode option in the Configuration menu is set to ON.
- The download device is set to the buffer.
- The baud rate is 2400 bps or less, or:
- The baud rate is 4800 bps or higher *and* hardware flow control is turned on.

Zmodem's streaming mode requires hardware flow control at higher speeds in order to keep up with the steady flow of incoming data. If hardware flow control is not turned on and the baud rate is at least 4800 bps, or if you are downloading to a disk device, Zmodem defaults to its regular "block" mode.

If you set the Allow streaming mode option in the Configuration menu to OFF, Zmodem will always default to "block" mode. You should set this if you have trouble downloading in streaming mode from a bulletin board or on-line service.

3.4.3.2 Ymodem batch

Ymodem is also a protocol that supports batch file transfers. It is an extension of the original Xmodem protocol (see below). It uses 1024-byte blocks to transfer data. It is slightly faster than Zmodem, but it has one disadvantage that makes Zmodem the preferred protocol.

The main inconvenience of Ymodem is that each block of data transferred must be of a certain length. If a file's size is not an exact multiple of this fixed block length, the end of the file will be "padded" with filler characters that were used to achieve the required block length in the last block. Novaterm can remove these fill characters automatically if you have the option turned on (see 4.7.9, **Chop X/Ymodem padding**), but there is risk of damaging the file when doing this, as explained in the referenced section.

3.4.3.3 Ymodem-g

Ymodem-g is an extension of Ymodem batch that operates in streaming mode. You may use this protocol *only* to upload or download directly to the buffer (because the Commodore 64 cannot simultaneously read characters from the modem and write them to a physical disk drive). There is no provision for error recovery in Ymodem-g; the first time an error is detected, the transfer is aborted.

Therefore, it is only suitable for downloading over modem connections with error correction, or across a null modem connection with another computer.

Ymodem-g holds the prize for being the fastest of the protocols, since there is no overhead in waiting for acknowledgments from the receiving computer. However, the lack of error recovery makes it impractical except over a modem link with error correction.

3.4.3.4 Xmodem-1k

This is an extension of the original Xmodem protocol that uses 1024-byte blocks. It can only transfer a single file at a time. Unlike Zmodem and Ymodem, both the receiver and sender must supply a file name, as the protocol will not transfer the name with the data. This protocol also has the “file padding” problem described above.

3.4.3.5 Xmodem-1k-g

Xmodem-1k-g is a streaming-mode extension of Xmodem-1k, just as Ymodem-g is the streaming-mode extension of Ymodem. Like Ymodem-g, you *must* upload or download directly to the buffer, and the transfer aborts at the first sign of an error. It is extremely fast, but must be used over a link with error correction.

3.4.3.6 Xmodem-CRC

This is the original Xmodem protocol. It supports both CRC checksums and additive checksums, so you may use this protocol even if the receiver does not support a CRC version of Xmodem. It transfers data in 128-byte blocks; file names are not transferred; and “file padding” does occur.

3.4.3.7 Punter (C1)

The Punter protocol (also called “C1”) was originally designed for use on Commodore computers, although it has fallen out of use in recent years. The Punter protocol can only transfer a single file at a time, but it does not suffer from “file padding”. It uses up to 255-byte blocks to transfer data. You're likely to find it only on Commodore-based bulletin board systems.

3.4.3.8 Multi-Punter

Punter multi-transfer is an extension to Punter that supports batch file transfers. It is available on a few Commodore-based bulletin board systems.

Note: This is *not* the same protocol as Punter's C2 batch protocol.

3.4.3.9 Kermit

The Kermit protocol is designed to work over communication links where other protocols aren't able to function for whatever reason. If control characters are intercepted or changed by a network switcher or other device before the receiving software sees them, file transfers with protocols like Xmodem and Ymodem will not work. Kermit seeks to overcome this problem by encoding all data characters as text. Because of this, and because it allows up to only 94 bytes per block, Kermit is very slow. It should only be used in situations where it is the only protocol that will work.

Kermit does support batch file transfers.

3.4.3.10 Wxmodem

This is a version of Xmodem designed for use over networks where the delay between sender and receiver is significant. It is designed to send up to 512 bytes of data at a time. It is provided in Novaterm for those situations where it is the only protocol available to the other

3.4.4 Selecting a block size

Some protocols allow you to vary the size of data blocks that are transmitted. Basically, a data block is a segment of the file being transferred. The data block is sent by the sending computer along with a checksum of the block. When the sender has transmitted a block, it then waits for the receiver to acknowledge it as either good or bad before continuing.

The size of that data block affects performance. With larger sizes, fewer acknowledgements must be sent by the receiver, so the file transfer is faster. However, if the communication link is noisy at all, many blocks may need to be retransmitted; therefore, a smaller block size will alleviate the overhead of having to retransmit large data blocks that were corrupted.

Only some of the protocols allow you to change the block size:

Zmodem allows block sizes between 32 and 1024 bytes.

Punter allows block sizes between 40 and 255 bytes.

Kermit allows block sizes between 20 and 94 bytes.

To change the block size, use the `Block size` option on the

Configuration menu (see 4.3.2, **Block size**).

3.4.5 Receiving a file

Once you have selected the protocol and requested one or more files from the BBS or on-line service, you are ready to initiate the download in Novaterm. To do this, press **C= D** from terminal mode (see 3.1.3, **Commodore key commands** , **C= D**).

Novaterm will display the file transfer screen. The window in the center of the screen displays data being received. Directly above the window, Novaterm maintains a count of the number of bytes received.

If you are not using a batch transfer protocol, Novaterm prompts you for a file name. Type in the file name as you wish it to be stored on the disk. If the file already exists on the disk, Novaterm will ask if you want to replace it.

If you are using a batch transfer protocol, and if a file name sent by the sender already exists on your disk, Novaterm changes the last character of the file name to avoid the conflict.

Novaterm will prompt you for a file type (unless you are using Punter or Multi-Punter protocol, which transfer the file type automatically).

Novaterm will then start sending signals to the other computer. The transfer should begin within several seconds. (If nothing happens after about 15 seconds, hold down the **C=** key to abort the transfer and try to find out what went wrong.)

At any time during the transfer, hold down the **C=** (logo) key to abort. You may have to hold it for a few of seconds before Novaterm gets to a point where it can abort the transfer smoothly.

3.4.6 Sending a file

Once you have selected the protocol and requested one or more files from the BBS or on-line service, you are ready to initiate the upload in Novaterm. To do this, press **C= U** from terminal mode (see 3.1.3, **Commodore key commands** , **C= U**).

Novaterm will display the file transfer screen. The window in the center of the screen displays data being sent. Directly above the window, Novaterm maintains a count of the number of bytes sent.

If you are not using a batch transfer protocol, Novaterm prompts you for a file name. Type in the file name as it exists on the disk.

If you are using a batch transfer protocol, Novaterm instead presents you with a file selection screen. (When Novaterm asks you for a **Pattern match**, enter any wildcard that will display a subset of files from the directory, or leave the entry blank to display all files.) In the left-hand column, Novaterm lists files from the disk directory. As you select files by highlighting them and pressing **RETURN**, Novaterm lists the selected files in the right-hand column. Pressing **F7** starts the upload.

Novaterm will then start sending signals to the other computer. The transfer should begin within several seconds. (If nothing happens after about 15 seconds, hold down the **C=** key to abort the transfer and try to find out what went wrong.)

At any time during the transfer, hold down the **C=** (**logo**) key to abort. You may have to hold it for a couple of seconds before Novaterm gets to a point where it can abort the transfer smoothly.

3.4.7 Sending and receiving summarized

The steps you must take to send files are summarized below:

For single-file protocols (**Xmodem-1k**, **Xmodem**, **WXmodem**, **Punter**):

1. Make sure you are set to the same protocol as the BBS or on-line service (see above).
2. Select the menu item or command on the BBS for "Upload file" or "Send file".
3. Tell the BBS the name of the file you want to upload.
4. Follow any instructions given by the BBS until it says it is ready to receive the file.
5. Then, press **C= U** from terminal mode.
6. Type in the name of the file as it appears on your disk.
7. At this point the screen should clear and file information displayed. The transfer should begin after a few seconds.

For batch transfer protocols (**Zmodem**, **Ymodem batch**, **Kermit**, **Multi-Punter**):

1. Follow the first five steps above. You may not have to tell the BBS or on-line service the names of the files you are going to upload.
2. Select the names of the files you want to upload, either by typing them in or by selecting them from the directory list. (See section 5.4, **Multiple file scratch** for a description of the file selection

screen.)

3. The screen should clear and the transfer should begin after a few seconds.

The steps you must take to receive files are summarized below:

For single-file protocols (**Xmodem-1k**, **Xmodem**, **WXmodem**, **Punter**):

1. Make sure you are set to the same protocol as the BBS or on-line service (see above).
2. Select the menu item or command on the BBS for "Download file" or "Receive file".
3. Tell the BBS the name of the file you want to download.
4. Follow any instructions given by the BBS until it says it is ready to send the file.
5. Then, press C= D from terminal mode.
6. Type in the name of the file as it will appear on your disk. If the file name you type is already on the disk, Novaterm asks if you want to replace it. Reply with *yes* or *no*.
7. Select the file type if requested.
8. At this point the screen should clear and file information displayed. The transfer should begin after a few seconds.

For batch transfer protocols (**Zmodem**, **Ymodem batch**, **Kermit**, **Multi-Punter**):

1. Follow steps one through five above, specifying multiple file names on the BBS instead of one file.
2. The screen clears and file information is displayed. The transfer should begin after a few seconds.

3.4.8 Transferring text files from non-Commodore systems

You may often find occasion to download text documents from non-Commodore-based on-line services. These files are always stored in standard ASCII text, which is not the same as Commodore PETSCII. If you were to transfer the file raw and try to read it on your Commodore 64, you would see that the uppercase and lowercase characters would seem to have switched places.

To correct this problem, Novaterm has a feature called "Transfer translation". When you turn this on, Novaterm performs ASCII translation on the file during the download before storing it to your disk. (It does the same kind of translation that occurs when you are in terminal mode. See 3.1.3, **Commodore key commands**, C= A.) To

toggle this feature, press **C= -** (minus) in terminal mode.

This feature also works when you upload a Commodore PETSCII text file to a non-Commodore-based service. Novaterm translates the file as it reads from your disk, so that the transmitted data blocks are ASCII-translated.

If you downloaded a text file without Transfer Translation, and later found it to be in standard ASCII, you can use the **ASCII file to PET** option on the Disk Operations menu to convert the file to Commodore PETSCII (see 5.3.5, **ASCII file to PET**).

3.4.9 Transferring files between two computers

Some people have (sadly) upgraded from the Commodore 64 to a newer computer. In the process, they often want to copy their valued documents from the C64 to the new computer. The easiest way to do this is to purchase a “null modem” or a “null modem cable”; you can find one at any computer store. This device connects two 9-pin or 25-pin RS232 serial ports together (they are usually available in both varieties). If your new computer is an IBM PC-compatible, it will have a 9-pin or 25-pin RS232 port. On the Commodore side, you will need either an RS232 interface (provides a 25-pin RS232 port from the user port) or a high-speed serial cartridge (see 1.1.3, **Specifying the serial device**).

To set up the computers for file transfers:

1. Connect the computers' serial ports with the null modem cable.
2. Start Novaterm on the Commodore 64.
3. Start any communications software on the other computer.
4. Set both computers to the same baud rate.

Now try typing characters on the Commodore 64 keyboard. You should see the characters appearing on the other computer screen. Make sure you also try typing characters on the other computer's keyboard; you should see them show up on the Commodore 64's screen. If characters in either direction are not appearing, check your connections.

Once the connection is working, you may transfer files. Make sure both computers are using the same file transfer protocol. To transfer files from the Commodore 64 to the other computer, press **C= U** in Novaterm's terminal mode, and select the “download” on from the other computer. To transfer files from the other computer to the Commodore 64, press **C= D** in Novaterm's terminal mode, and select

the “upload” option on the other computer.

4. CONFIGURATION

The Configuration menu allows you to modify various settings for Novaterm. The options cover the whole range of Novaterm's operations. Many of the options may also be set elsewhere in the program. The Configuration menu, however, is where you'll find all configuration options in one place.

Most of these options may be saved in a configuration file that is loaded each time you start Novaterm (see 4.5.2, **Save configuration**). Using a configuration file is highly recommended. In fact, if you followed Novaterm's initial configuration procedure (see 1.2, **Starting up the program**), a configuration file was created for you.

The Configuration menu is split into two halves, because there are more options than will fit on the screen at one time. To switch between the two parts of the Configuration menu, use the cursor-right key.

Options on the configuration menu are physically grouped according to the category of the option. This section describes each category and option in detail.

4.1 Communication options

4.1.1 Terminal emulation

This option allows you to change the terminal emulation (see 3.1.1, **Terminal emulations** for a description of terminal emulations). It has the same effect as pressing C= T in terminal mode (see 3.1.3, **Commodore key commands**).

4.1.2 Baud rate

The baud rate is the speed at which data is sent and received from the modem. (Actually, the more accurate term for this option is the "BPS rate", where BPS stands for "bits per second". "Baud rate" is not the same thing. However, for purposes of clarity among users of the software, the terms "baud rate" and "BPS rate" are used interchangeably in Novaterm.)

Novaterm can operate at 2400 baud through the user port, and higher using an enhanced serial device, such as the SwiftLink cartridge. The highest speed you can select here will be limited by the serial driver and modem driver you have installed. (Each serial driver and modem driver stores the maximum baud rate that may be used with it.)

If you have a 14.4k or 28.8kbps modem, you will notice that there are no settings for those speeds. This is because high speed modems support data compression and error correction. 14.4kbps or 28.8kbps indicates the rate of *compressed, error-corrected* data transfer. The actual rate of "real" data (before and after compression) will be at least 14.4k/28.8k, but usually higher. That's what your modem box meant when it advertised something like "speeds up to 57.6k possible!" Therefore, these modems allow you to set your terminal program to a speed higher than the modem's rated speed, in order to take advantage of compression and error correction.

The standard way to do this is to always set Novaterm to the highest possible baud rate, and turn on hardware flow control. Hardware flow control allows Novaterm and the modem to dynamically manage the flow of data depending on how well the compression is performing (which depends on the type of data being transferred). (see 4.6.2, **Flow control**)

4.1.3 Word/parity

Most on-line services operate at 8 bit word length and no parity. Therefore, you should set this option to read 8N1. Some on-line services may require you to use "even" or "odd" parity instead, in which case this setting should be 7E1 or 7O1. However, you should specify the `Word/parity` setting in the dial entry for each on-line service in your dialing directory, since the setting can be different from one service to the next. It's best to leave the setting on the Configuration menu set to 8N1, and just let it change depending on the service you dial.

4.2 40/80-column settings

4.2.1 80-column driver

This option specifies the driver to use for 80-column mode. In other words, this option determines which set of programming code to use to display the 80-column screen. The drivers currently available are:

default: The standard, unassisted 80-column driver that uses the C64's high-resolution bitmapped screen.

fast-REU: Uses a RAM Expansion Unit (1700/1750/1764) to implement a fast scroll on the C64's hi-res screen. You must use the REU memory driver in conjunction with this driver (see 1.1.7, **Specifying the memory device**).

VDC (25): Uses the C128's 80-column screen in 25-line mode. Use this driver only if you are running Novaterm on a C128 in 64 mode, and you have an RGB 80-column monitor. This driver, as well as the other VDC drivers, uses the C128's 2 MHz fast mode.

VDC (28): Uses the C128's 80-column screen in 28-line mode. You may need to adjust your monitor to see all 28 lines.

VDC (43): Uses the C128's 80 columns in 43-line interlaced mode.

VDC (50): Uses the C128's 80 columns in 50-line interlaced mode. You may need to adjust your monitor to see all 50 lines.

4.2.2 40-column font

This option specifies the font to use in 40-column mode.

4.2.3 40-column color

This option lets you change the color scheme used in 40-column mode. You may change the text, background, and border colors, as well as the color of the status line used in 40-column terminal mode.

4.2.4 80-column color

This option lets you change the color scheme used in 80-column mode. Since 80-column mode can be hard to read, you should experiment with different text and background colors to find a combination that is easiest for your eyes, if you are not happy with the defaults (see 3.1.6.1, *Improving readability*).

4.3 Modem setting options

4.3.1 Modem type

This option changes the modem driver currently installed. Choose the driver that corresponds to your modem:

- 1650:** Commodore 1650 modem (300 baud).
- 1660:** Commodore 1660 modem (300 baud).
- Hayes 1200:** Hayes-compatible 1200 baud (and 1670 modem).
- Hayes 2400:** Hayes-compatible 2400 baud modem.
- Hayes 9600:** Hayes-compatible 9600 baud modem.
- Hayes fast:** Hayes-compatible 14.4k or 28.8k modem.

4.3.2 Modem init

The modem initialization string is used only when you have installed

one of the Hayes-compatible modem drivers. It is a sequence of Hayes AT commands that is sent to your modem when Novaterm starts up. In other words, Novaterm configures your Hayes-compatible modem at start-up using this series of commands. The initialization string may also be sent manually by pressing C= I in terminal mode (see 3.1.3, **Commodore key commands**, C= I).

In the Configuration menu, Novaterm does not display the entire initialization string; it cuts off the string to avoid a strange display. However, the entire string is still there, even though you can't see it all. The string may be up to 50 characters long.

Sending Hayes commands to a modem is like flipping switches inside the modem to make it behave in a desired way. The initialization string consists of a series of these commands bunched together so that they all take effect at once. Because of this, it can be difficult for the untrained eye to discern individual commands in the string. A command begins with either a single letter, an ampersand (“&”) followed by a single letter, or a percent (“%”) followed by a single letter. Most commands have a one- or two-digit number following the letter, and a special class of commands known as the “S” registers also have an equal (“=”) sign followed by a second number.

Some of the more common commands are listed below. Your modem's manual contains more complete descriptions of these commands. Some of these commands may not be supported by your modem if it is an older or slower model.

- E1 - Turn on modem's echo in command mode.
- M1 - Turn modem speaker on until connected.
- V1 - Turn on “verbose” reporting from the modem.
- X4 - Report all status codes.
- &C1 - Make carrier detect work properly. (This is very important)
- &D2 - Hangup when DTR is dropped. (Novaterm likes this also)
- &K3 - Hardware flow control. (required with high-speed modems)
- S0=0 - Turn off auto-answer.

4.3.3 Dial method

The dialing method may be either tone dial or pulse dial. Pulse dialing is like the old rotary phone method of dialing. It may be necessary if you live in an area where tone dialing is not supported (in the United States, tone dialing should be available everywhere).

4.3.4 Redial time

When redialing a number from the dial menu, this option specifies the number of seconds to wait before retrying the number.

4.3.5 Dial pacing

A large number here slows down the rate at which digits are dialed on a Hayes-compatible modem. If your modem has trouble dialing, it may be because the digits are being sent too fast; try increasing this.

4.3.6 Carrier type

This option applies only when using the `User port serial driver`. It specifies whether the carrier detect line from the modem indicates a carrier when high (`Normal`) or when low (`Inverted`). (The “carrier detect” indicates whether there is currently a connection established with another modem.) This must be set properly, or Novaterm features that depend on a proper carrier detect will not function, such as redialing and the on-line timer.

To set this, make sure your modem is hooked up and turned on, but is not on-line. Go to terminal mode and check the state of the indicator letter `C` in the status line. If the `C` is reversed (highlighted) even though there is no connection, change the carrier type to the opposite of its current setting.

4.4 File transfer options

4.4.1 Protocol

This option lets you change the installed protocol. It has the same effect as pressing `C= P` in terminal mode. (See 3.4.3, **Selecting a protocol** for more information.)

4.4.2 Block size

This option allows you to change the block size for the currently installed protocol (see 3.4.4, **Selecting a block size**).

4.4.3 Chop X/Ymodem padding

With this option turned on, Novaterm automatically removes any “padding” added by the Xmodem and Ymodem family of file transfer protocols. (See 3.4.3.2, **Ymodem batch** about file padding.)

4.4.4 Zmodem auto-download

This option controls whether the Zmodem auto-download feature is to be used. If you do not have the Zmodem protocol installed, this

option has no effect (see 3.4.3.1, **Zmodem**).

4.4.5 Allow Zmodem streaming

With this option turned on, Zmodem will operate in “streaming” mode if the following conditions are met:

- The download device is set to the buffer.
- The baud rate is 2400 bps or less, or:
- The baud rate is 4800 bps or higher *and* hardware flow control is turned on.

Turning off this option disables streaming mode in all cases. You should do this if you have trouble downloading from a particular service in streaming mode. (See section 3.4.3.1, **Zmodem** for more information about Zmodem's streaming mode.)

4.4.6 Existing files policy

This setting determines what happens during a batch download when a file to be downloaded already exists on disk.

- **Replace:** Deletes the file on disk and creates a new one with the same name.
- **Rename:** Renames the incoming file by incrementing the last character of the file name.
- **Resume:** If the Zmodem protocol is selected, Zmodem attempts to resume transfer of the file. (See 3.4.3.1, **Zmodem**) If a different protocol is selected, the incoming file name is renamed.

4.5 Configuration file options

4.5.1 Load configuration

This option re-loads the configuration file you previously saved, reverting all options to their original startup settings.

4.5.2 Save configuration

This option saves all of your current settings to a file. This file is read by Novaterm when you start the program, and the options are all set for you at that time.

4.6 Serial device settings

4.6.1 Serial driver

Select the driver corresponding to your serial communications

hardware. See section 1.1.3, **Specifying the serial device** for a description of each driver.

4.6.2 Flow control

Flow control allows Novaterm to keep up with incoming data. When Novaterm falls behind in processing incoming data, Novaterm tells the remote computer to pause until Novaterm catches up. When Novaterm is nearly caught up, it tells the remote computer to resume transmission. This prevents characters from being lost. Flow control is done automatically, but it may be turned off by toggling this option.

Novaterm supports two types of flow control:

Software (XOFF/XON): In software flow control, Novaterm sends an XOFF character (usually CTRL-S) to pause and an XON character (usually CTRL-Q) to resume.

Hardware (RTS/CTS): In hardware flow control, Novaterm tells your modem to stop delivering data to the computer by lowering the RTS line (one of the RS232 port pins). In this state, the modem stores incoming data in a buffer until Novaterm raises the RTS line again. In addition, the modem can tell Novaterm to stop giving it data by lowering the CTS line (which can happen if the remote computer has lowered its RTS line). Hardware flow control is much more precise and reliable than software, since modems respond immediately to changes in RS232 lines. However, to use hardware flow control, you *must* have at least a 9600 bps modem. Most slower modems (2400 bps and below) do not respond to the RTS and CTS lines, so hardware flow control is not possible with them.

If you are using a high speed modem (9600, 14.4k, or 28.8k) you *must* use hardware flow control to take advantage of data compression and error correction, and to enable Novaterm to keep up with incoming data (see 4.1.2, **Baud rate**).

At speeds of 2400 bps and below, flow control is not necessary, and it can be set to `None`.

4.6.3 Flow tolerance

The "flow tolerance" specifies the number of characters that may build up in the receive buffer before Novaterm stops the flow (using either hardware or software flow control). Since the receive buffer holds a maximum of 256 characters, the tolerance may be any number between 6 and 256. The default setting is 220, which is effective for

speeds up to 38,400 bps using hardware flow control. (If you are using a HART cartridge at 57,600 bps, you may need to reduce this setting to avoid losing data.)

If you are using software flow control, this value should be much lower, since there is often a significant delay between when Novaterm sends the XOFF signal and when the remote computer actually stops sending data.

4.6.4 XOFF character

With software flow control, this is the ASCII value of the character used to stop the flow of incoming data. The default is 19, or CTRL-S, which is the standard key used for pausing the flow of data. Some bulletin boards use a different key for pausing, so you would need to change this in that situation, if you were using software flow control.

4.6.5 XON character

With software flow control, this is the ASCII value of the character used to resume the flow of data. The default is 17, or CTRL-Q, which is the standard key used for this purpose. Some bulletin boards use a different key for this, so you would need to change this option in that situation, if you were using software flow control.

4.6.6 Line pacing

This option specifies the number of seconds to wait between lines when transmitting a text file with the `Textfile` to modem or `Buffer file` to modem options.

4.7 Miscellaneous options

4.7.1 Define F-keys

This option brings up the function key definition menu. (See 3.1.4, **Function keys** for information on how these keys work.) At the function key menu, either press a function key or select it on the menu to edit the key's definition.

Control keys may be imbedded within a function key string by including a caret symbol ^ followed by the control key letter. For example, to include the control-B in the string HELLO [CTRL-B]THERE, you would type HELLO ^BTHERE. A carriage return (which is CTRL-M) could be placed at the end of a string by entering HI THERE^M.

A 1/2 second pause in a string is represented by a CTRL-P. For example, to define a function key to send two strings delimited by carriage returns with a full second pause in between, the string might look like: SEND ME^M^P^PTO THE MOON^M. Pressing the function key containing this string will display SEND ME with a return, wait one second, and display TO THE MOON followed by another return.

Function key definitions may be saved in individual files by selecting Save function keys on the same menu. In order to tell Novaterm to reload the same function key definition file each time you boot the program, you must also save the configuration file (see 4.5.2, Save configuration), which will record the current function key file name.

4.7.2 Device settings

Novaterm allows you to specify which device number to use for various operations. This option brings up the menu that lets you define these devices. (This menu can also be brought up from the disk functions menu and the main menu, for the sake of convenience.)

4.7.2.1 Device setting categories

Novaterm has four categories of operations where disk access is involved. You may specify a disk device for each of these operations:

Program device: Novaterm uses this device to load modules, the configuration file, phone books, function key files, and script files. This should be the disk from which you originally loaded Novaterm.

Upload device : When you upload files, Novaterm looks to this drive.

Download device : Novaterm saves downloaded files on this drive.

Buffer device: Novaterm uses this device to load or save files to disk from the buffer. The automatic buffer save uses this device (see 4.7.8, Save buffer when full).

To set the disk device number for any of these options, enter the device number, followed by a comma, followed by the drive/partition number. The drive/partition number (distinct from the device number) is used when you have a single disk device containing two or more drives, or a partitioned hard drive or RAMLink/RAMDrive. If you are using only single disk drives, the drive number should always be 0.

Novaterm accepts values between 8 and 30 for disk device numbers. (Novaterm also accepts a disk device of 1, which is used by some

enhancement hardware as a special device.) When you enter a device number for any of the above options, Novaterm checks to see if that device is actually connected. If it is not, the device number is not changed.

For the Program device, Upload device, and Download device, typing a letter B in place of the device number directs these functions to Novaterm's buffer. In this case, the drive/partition value specifies the buffer subdirectory number to use.

4.7.2.2 Associated disk commands

Each option may also have a disk command associated with it. These commands are listed to the right of the device numbers. Novaterm always executes the corresponding disk command before switching to the device. For instance, if you had the disk command for the Download device set to i0, Novaterm would send the i0 command to that device (which would initialize the drive) before downloading a file.

The most practical use for these associated disk commands is to allow operations to use different subdirectories or partitions on a single drive that allowed it, such as a 1581, a RAMLink/RAMDrive, or a CMD hard drive.

For instance, suppose you have a RAMLink/RAMDrive which occupies device 16, and you have four subdirectories set aside for Novaterm on partition #2:

```
//novaterm
//novaterm/uploads
//novaterm/downloads
//novaterm/buffer
```

You would like to have uploads come from the uploads directory, downloads go to the downloads directory, and buffer files saved to the directory called buffer. To set this up, you would specify change-directory commands for the associated disk commands. The device settings would appear as follows:

```
Program drv   : 16,2 >cd//novaterm
Upload drv    : 16,2 >cd//novaterm/uploads
Download drv  : 16,2 >cd//novaterm/downloads
Buffer drv    : 16,2 >cd//novaterm/buffer
Printer dev   : 4,7
```

4.7.2.3 Printer device

For the Printer device, the second number is the secondary address of the printer. Most serial printers require a secondary address of 7 for lowercase text. If you have problems getting your printouts to look right, check your printer manual to see what secondary address is recommended.

Novaterm accepts values between 4 and 30 for the printer. Novaterm does not change the printer device if the device number you specify is not connected.

4.7.3 Phone book

This specifies the file name of the phone book currently in use. Changing this option on the configuration menu does not load a new phone book; it merely changes the name of the current one.

4.7.4 F-key file

This specifies the file name of the function key definition file currently in use. Changing this option on the configuration menu does not load a new definition file; it merely changes the name of the current one.

4.7.5 Startup script

This option specifies the file name of a script file that is executed automatically when you start up Novaterm (see section 6, **Scripts**).

4.7.6 Memory device

This option lets you change the memory device driver currently installed. The memory device driver is used by the buffer to store data. This can also be changed from the Buffer Menu (see 1.1.7, **Specifying the memory device**).

4.7.7 Set the time

This option lets you select a real-time clock device, and sets the system clock from it. See section 1.1.10, **Selecting a real-time clock device** for more information.

4.7.8 Print in ASCII

If this option is turned on, Novaterm converts all text sent to the printer to standard ASCII. Turn on this option if you have an ASCII printer. This option alone will not print linefeeds at the end of each line; use the following option for that.

4.7.9 Print with linefeeds

If this option is turned on, Novaterm adds a linefeed to every carriage return sent to the printer. If your printer is printing each line on top of the last without advancing the paper, turn on this option.

4.7.10 Save buffer when full

If you do not have a memory expansion device, you are limited in the amount of buffer space available. With a stock Commodore 64, you have only 5.5K of memory for the buffer.

To aid in capturing a large amount of text with a limited buffer space, Novaterm has the ability to periodically save the buffer to disk and clear it out when it fills up. This option turns the feature on or off.

When the buffer becomes full during a text capture, Novaterm immediately tries to pause the flow of data. How it does this depends on the type of flow control being used (see 4.6.2, **Flow control**). Once the host computer stops sending data, Novaterm saves the capture file to disk, appending it to the disk file if it already exists. The capture file in the buffer is erased and re-opened, making all the memory available again. Then, Novaterm requests to resume the flow of data (again, depending on the type of flow control). The buffer capture continues after this, and the process repeats when the buffer fills up again. When the text you want to capture has all been received, you must manually save the remaining text. Novaterm allows you to append this text to the existing disk file. The result is the equivalent of one large capture file saved to disk.

If a disk error occurs while trying to save the capture file, Novaterm does not resume transmission. This leaves the remote system paused so that you may correct the problem, save the capture file manually, and resume the text capture where it left off.

If you are using a memory expansion device, and thus any memory device driver other than `internal`, you should turn off this option, since you have a great deal of space available for capturing text.

4.8 Notes on changing hardware and drivers

If you have more than one serial device and/or memory expansion device, you may need to take some extra steps when physically changing hardware and changing Novaterm's drivers.

If you attempt to load a driver for one device that uses the same registers as another device, Novaterm may lock up. For instance, the

HART cartridge and the BBGRam cannot coexist because they share some common I/O registers. Suppose you had HART cartridge plugged in but not the BBGRam, and you had the HART serial driver and internal memory driver installed. Then, you decide to unplug the HART and put the BBGRam in. You do so, then load up Novaterm again. It locks up! This is because Novaterm is still loading the HART driver as specified in the configuration file. When it loads the HART driver, the function that tests for the presence of the cartridge interferes with the BBGRam, and the system locks up (or does something else unpredictable). To avoid these conflicts:

1. Unplug *everything*.
2. Load up Novaterm.
3. Change drivers and *save the configuration*. (Novaterm will not lock up if a device it is looking for simply isn't present.)
4. Plug in the new hardware.
5. Start Novaterm again.

As you get used to switching things around, you'll eventually determine which cross-configured drivers won't lock up the system. The I/O registers used by common serial and memory devices are listed below. A device doesn't necessarily use all registers it reserves, but it occupies more address space due to "mirroring" of registers.

Memory drivers

internal:	No I/O registers used.
VDC+internal:	Uses \$D600-\$D601.
REU:	Uses \$DF00-\$DF0A, mirrored every 32 bytes.
BBG/GEORam:	Uses \$DE00-\$DEFF and \$DF80-\$DFFF.
RAMLink:	Uses \$DE00-\$DEFF and \$DF10-\$DFFF.
RAMDrive:	Uses \$DE00-\$DEFF and \$DF10-\$DFFF.

Serial drivers

User port:	CIA #2 registers used.
UP9600:	CIA #1 and #2 registers used.
SwiftLink:	Uses \$DE00-\$DEFF (\$DE00-0F with AddressFixer).
CommPort:	Uses \$DF00-\$DFFF (\$DF00-0F with AddressFixer).
SL-DE20:	Uses \$DE20-\$DE2F (with AddressFixer).
SL-DF20:	Uses \$DF20-\$DF2F (with AddressFixer).
SL-D700:	Uses \$D700-\$D7FF (\$D700-0F with AddressFixer).
HART:	Uses \$DE18-\$DE1F and \$DF18-\$DF1F.

5. DISK UTILITIES

Novaterm provides a menu of options to perform common disk operations. This section covers the options listed on the Disk Operations menu.

Note: All disk operations use the “current” disk device. The current device is displayed at the top of the menu. The current device number may be changed using the “Disk command” option (see below).

5.1 Disk command

You may send a disk command to the current device by selecting `Disk command`. In addition to the normal DOS commands, Novaterm supports a few others:

Directory: The disk directory may be called up by entering a `$` symbol. Pattern matches are also allowed. For example, to get a directory of files whose names start with `prt`, you would type: `$:prt.*`

Read error channel: Type `@` by itself to read the drive's error channel.

Select device: The `#` symbol selects a different disk drive. For example, to switch to device 9, drive 0, enter: `#9,0`

5.2 Disk directory

This option lists the directory of the disk. Pressing the `P` key pauses the listing; pressing the `S` key stops it.

5.3 Textfile (SEQ file) options

Textfiles on the C64 are labeled as `seq` files in a disk directory. In the following discussion, “textfiles” and “seq files” have the same meaning.

5.3.1 View textfile

This option lists a textfile on the screen.

5.3.2 View textfile in 80 columns

This option lists a textfile on the screen using 80-column mode.

5.3.3 Print textfile

This option prints a hard copy of the file.

5.3.4 Textfile to modem

This option outputs a textfile line by line to the modem. (This is sometimes called an "ASCII upload".)

For instance, if you wanted to enter a message on a BBS that you had already typed and saved in a `seq` file using Novaterm's text editor, you would use this option. On the BBS, you would execute the command to enter a message. When the BBS's message editor is ready, you would go to the Disk Operations menu, select `Textfile to modem`, and give the name of the file you created previously.

As Novaterm sends the file one line at a time, the BBS will echo the characters back to Novaterm, just as if you were typing them. There is often a delay involved in this which you do not notice when typing, but which will become significant since Novaterm sends the data as fast as possible. To compensate for the delay, Novaterm must be told to wait a certain number of seconds between each line to give the BBS time to catch up. This number of seconds is specified with the `Line pacing` option in Novaterm's Configuration menu (see 4.6.6, **Line pacing**). A line pacing of zero eliminates the delay completely.

5.3.5 ASCII file to PET

This option converts a textfile from standard ASCII to Commodore PETSCII. (See 3.3.7.5, **ASCII to PET**)

5.3.6 PET file to ASCII

This option converts a textfile from Commodore PETSCII to standard ASCII, ending each line with a carriage return/linefeed. (See 3.3.7.6, **PET to ASCII**)

5.3.7 PET file to Unix

This option converts a textfile from Commodore PETSCII to standard ASCII, ending each line with only a linefeed. (See 3.3.7.7, **PET to Unix**)

5.3.8 UUdecode file

UUdecodes a file. (See 3.3.7.8, **UUdecode file**)

5.3.9 UUencode file

UUencodes a file. (See 3.3.7.9, **UUencode file**)

5.3.10 UUencode file in PET

UUencodes a file in PETSCII. (See 3.3.7.10, UUencode in PET)

5.4 Multiple file scratch

This option brings up the file selection screen and allows you to select files to be scratched. Before bringing up a menu of files, Novaterm prompts you for a pattern match. This allows you to enter Commodore DOS wildcards to bring up only certain files. For example, if you entered `per*`, only file names beginning with the letters `per` are shown. Pressing RETURN here without an entry brings up all files.

The file selection menu brings up the first 21 files on the disk and displays them in the left-hand column. To select files, move the menu bar to the file you want and press RETURN or F3. This puts the file name in the right-hand column. Repeat this until you have selected all of the files you want. To call up the next 21 files from the directory, press F5. If you select a file you don't want, move the menu bar to the file name in the right-hand column and press F4 to remove it. Once you are finished, press F7 to start erasing the files you have selected.

With the menu bar in the right-hand column, you may edit or add file names by typing them in directly rather than paging through the directory list. Move the menu bar to the file name you wish to edit, and press RETURN. You may then re-type the file name. If you move the menu bar to the blank line below the last file name and press RETURN, you may type in a new file name.

6. SCRIPTS

Scripts are a powerful tool for making Novaterm perform functions automatically. Scripts can do everything from load modules to download files to respond to specific text received from an on-line service. Scripts can log you into a service, collect electronic messages, download files, and do a variety of other things, all unattended.

You write a script using Novaterm's script language. The language is just a set of one-line instructions that Novaterm recognizes to perform different functions. A script consists of a series of these one-line instructions. Novaterm executes each instruction in order. When you write a script, you are just telling Novaterm what to do, one step at a time.

To actually create a script and run it, you'll follow these steps:

1. Load Novaterm's Text editor program from the Utility Modules menu.
2. Type in your script, line by line.
3. Save the script *source* to a file. The "source" is the text you just typed. This isn't the actual script yet...
4. Compile the script. "Compiling the script" means to convert the text you typed into an encoded form that Novaterm can recognize. You compile the script by pressing the RUN/STOP key in the Text Editor, while your source is there on the screen.
5. Return to Novaterm, and execute the script by pressing C= G in Terminal Mode (see 3.1.3, **Commodore key commands**, C= G).

Each of these steps is described in more detail below.

6.1 Creating a sample script

To illustrate how scripts work, let's create a sample script and execute it.

6.1.1 Determining the sequence of responses

Suppose we want to create a script that automatically logs us into our Internet provider, a bulletin board called Eskimo North. To do this, the script must watch the text sent by Eskimo North as it comes in, and output our login name and password at the right time.

Novaterm 9.6 User's Guide

To set up our script to do this, we must examine the text that displays during a normal login procedure (i.e. when we dial and log in manually). To do this, we can capture the text in the buffer and look at it later.

Listed below is a sample buffer capture of the text sent from Eskimo North during the login procedure. Any text that we had to type, in response to a prompt, is shown in bold.

SAMPLE LOGIN TO ESKIMO NORTH

Eskimo North Connection Selection

- 0) Apply for a 2-week free trial
- 1) Connect to Eskimo [general login sessions]
- 2) Connect to Eskinews [uqwk uucp and mud maintenance]
- 3) Connect to Isumataq [irc bots only]
- 4) Connect to Tial [use this host for TIA slip]
- 5) Connect to Mail [use for UUCP mail connections]
- 8) Western Washington BBS List
- 9) Exit

Your Selection ==> **1**

New Users - please type "new".

Type your login in lower case (small letters).
Passwords are case-sensitive THISPW != thispw.
Passwords will not echo.

login: **voyager**

Password: *********

Last login: Mon Jun 26 13:55:30 from tial.eskimo.com
SunOS Release 4.1.4 #2: Tue Jun 27 04:07:39 PDT 1996
North Everett : (206) 258-0759 300 bps - 14.4kbps
South Everett : (206) 742-1150 300 bps - 14.4kbps
Seattle/Maple Villy: (206) 367-3837 300 bps - 14.4kbps
Des Moines / Kent : (206) 838-9513 300 bps - 14.4kbps
Tacoma / Auburn : (206) 927-3218 300 bps - 14.4kbps
East Side Line : (206) 451-8876 300 bps - 14.4kbps
Enumclaw/Blk Dmnd : (206) 939-5992 300 bps - 14.4kbps
Buckley : (360) 802-0992 300 bps - 14.4kbps

Eskimo North Users Meeting: Sunday July 20th 2:30PM @
Ballard Godfathers (63rd Street and 15th Ave N.W.)

You have new mail.

Main Command?

END OF SAMPLE LOGIN

Notice that there were three places we had to type in a response:

1. In response to the string `Your Selection ==>` we typed a 1 and pressed RETURN.
2. In response to the string `login:`, we typed the user name `voyager` and pressed RETURN.
3. In response to the string `Password:`, we typed our password and pressed RETURN.

After responding to those three prompts, we were logged in. Therefore, we want our script to automatically respond to these three prompts for us.

After responding to the prompts, we want the script to end. To do this, we'll find a fourth string that, when detected, ends the script. The string `Command?`, at the end of the sample capture file, is Eskimo North's command prompt, which is where we want to take back control. Therefore, in response to the string `Command?`, we'll end the script.

Now that we know what to look for, let's start writing the script. First of all, let's have our script monitor the incoming text for the four strings we identified above. In a Novaterm script, we may specify up to six strings to search for at one time, and then we put the script into a loop that enters terminal mode and waits until one of those strings is detected.

To set up our four search strings, we use the `check script` command. The next four lines of our script are:

```
check 1 "Your Selection ==>"
check 2 "login:"
check 3 "Password:"
check 4 "Command?"
```

The `check` command takes two arguments. The first is a number between one and six; this identifies an index number for the string. The second argument is the string itself. You must put quotation marks around the string to identify it.

The above four commands merely set up our search strings. Now, we must put Novaterm into a loop and wait for the first occurrence of one of the strings. To do this, the following line of the script must be:

```
wait
```

The `wait` command puts Novaterm into terminal mode. Novaterm displays incoming text until one of the search strings is detected. The script will *not* continue beyond this line until one of the strings is

found.

When one of the strings is found, the script continues on to the next line. Therefore, our next few lines must figure out which of the strings was detected, and respond to it appropriately.

First, let's check to see if the string `Your Selection ==>` was found:

```
if check 1 then output "1^m"
```

There's a lot to this statement, so let's look at it in pieces.

The `if` command is a conditional statement. It checks to see if the specified condition is true, and if it is, the command following the word `then` is executed. If the condition is false, the command following `then` is not executed, and the script continues on to the next line.

In this case, the condition `check 1` is telling Novaterm to check if string number 1 (`Your Selection ==>`) was detected. If it was, the script should execute the command:

```
output "1^m"
```

The output command simply sends the specified string (which must be surrounded by quotation marks) to the modem. In this case, the string consists of the character 1 and a RETURN. (The sequence `^m` represents the equivalent of the RETURN key, or CTRL-M. This notation is identical to the way control characters may be embedded in function key definitions. See 3.1.4, **Function keys** for a detailed explanation of this.)

Now, since we don't know which string was detected by the `wait` command, we must include conditions for the other two strings:

```
if check 2 then output "voyager^m"  
if check 3 then macro 1  
if check 4 then end
```

After the `wait` command exits, if the detected string was not string #1, the script falls through to the above statements. If the detected string was string #2, the login prompt, the script responds with our login name, followed by a RETURN.

If the detected string was string #3, the password prompt, the script executes the `macro` command. The `macro` command outputs the contents of the specified function key, as it is defined in the `Edit`

Function Keys menu. The command `macro 1` outputs the contents of function key #1, or F1. Remember that the F1 macro key is automatically defined with the password of the on-line service during the dialing process. If we have our password specified in the dialing entry for Eskimo North, then `macro 1` will output that password.

If the detected string was string #4, the script executes the `end` command, which ends the script.

The last line of our script must re-enter the loop to search for the remaining strings if the fourth string was not found. To do this, we will first *modify* the line with the `wait` command to read:

```
loop wait
```

The word `loop` is not a script command; it is a label that functions as a placeholder at a certain line in the script. We can now use the label `loop` in our last line:

```
jump loop
```

The `jump` command tells the script to continue execution at the line identified by the label `loop`. In our script, this command jumps back to the `wait` command, continuing the search process. Note that if string #4 is detected, the script ends (by way of the previous line, `if check 4 then end`) before it can get to the `jump` command.

Let's do one more thing. Suppose we want to make sure Eskimo North is going to respond to us after we connect. Sometimes, you have to press the RETURN key a couple of times to get an on-line service to respond. To cover this contingency, we insert the following line at the beginning of our script:

```
output "^m"
```

This sends a carriage return before doing anything else.

Let's list out the entire script:

START OF SCRIPT

```
; Novaterm 9.6 auto-login script for Eskimo North
; (comments may be inserted in the script by placing a
; semicolon before them)
output "^m" ; force Eskimo to respond
check 1 "Your Selection ==>" ; set up four strings
check 2 "login:"
check 3 "Password:"
check 4 "Command?"
loop wait ; wait in terminal mode
if check 1 then output "1^m" ; respond to "Selection"
if check 2 then output "voyager^m" ; respond with user name
if check 3 then macro 1 ; respond with password
if check 4 then end ; exit if "Command?" found,
jump loop ; otherwise continue
```

END OF SCRIPT

6.1.2 Typing in the script

Now that we know our script line by line, we can type it in and compile it. To do this, we'll run the Text Editor program by selecting Text editor from Novaterm's Utility Modules menu.

If you are unfamiliar with how the text editor works, see Appendix A, **Text Editor** for a description.

First, we'll type in the script just as it appears above. It's very important that you use lowercase letters for all of the script commands. Strings that are enclosed in quotes (such as the search strings) may contain both upper and lowercase letters. (To save time in typing the script, you may leave out all of the comments.)

Now, we'll save the script we just typed to a text file so we can look at or change it later. Use the CTRL-S command, and choose a file name. This script could be called `eskimo.ss`. I use the extension `.ss` to denote script source files (the "ss" stands for "script source").

It's important to understand that the file you just saved is the "source code" for the script, not the actual script that Novaterm can execute. To convert this source code into a form that Novaterm can recognize, we must first "compile" the script. This conversion process, or compilation, is built into the Text Editor. To do this, press RUN/STOP in the Text Editor (it won't break out of the program!). The Text Editor compiles the script quickly, and asks you for a file name for the compiled script. The compiled script could be called `eskimo.sc`. I use the extension `.sc` to denote executable scripts (the "sc" stands for

“script compiled”).

6.1.3 Executing the script

Since this is a script that must be executed *after* we are connected to Eskimo North (after all, checking for the `login:` prompt won't do much good if we aren't connected), we want to specify this script as the “Connect script” for Eskimo North. If you edit the dial entry for Eskimo North from the dialing directory menu, you'll see an item for specifying the `Connect script`. Select this item, and type in the name of the *compiled* script, `eskimo.sc`.

Now, when we dial Eskimo North from the dialing directory, our new script is executed immediately after the connection is established. If all goes well, you'll see Novaterm automatically respond to each of the prompts we identified.

6.1.4 A slight variation

In the above example, we had to dial Eskimo North ourselves from the dialing directory to execute our script. Suppose now we want our script to do the dialing for us. We can insert the following line at the beginning of our script (after the `.opt ansi` command):

```
dial "Eskimo North"
```

The `dial` command searches our dialing directory for Eskimo North and dials it. The rest of the script continues after the connection is made.

Now, instead of putting our script in the `Connect script` field for Eskimo North, we can execute the script directly. Press `C= G` in Terminal Mode, and type in the name of the script, `eskimo.sc`. The script will now take care of the dialing for us.

6.1.5 Starting Novaterm with the script

Suppose we use this script so often that we would like to have it execute automatically when we start up Novaterm. Each time we load up Novaterm, the script would automatically dial Eskimo North and log us in, without any help from us!

To do this, select `Configuration` from the main menu, press the cursor-right key, and then select `Startup script`. Type in the name of the script, `eskimo.sc`. Then, save your configuration file by selecting `Save configuration` so that the script will be executed the next time Novaterm is started.

6.2 Script commands by category

Listed below is the entire set of Novaterm script commands. Remember that the commands must be typed in *lowercase*! If script commands are typed in uppercase, *they will not compile correctly*.

Any arguments that go along with a command are placed next to it. Numeric arguments are given in brackets [], and string arguments are in quotes " ". Optional arguments are given inside < and > symbols. The description of the argument is in lowercase.

6.2.1 Modem and screen input/output

```
check [string number] "string"
```

Works in conjunction with the wait command to detect incoming strings. For example,

```
check 2 "Password"
```

defines string number 2 as Password. Up to 6 strings may be defined. When the script encounters a WAIT command, it goes to terminal mode and wait until one of these defined strings is received. The ordering of these commands in a script should be as follows:

```
check 1 "Username"  
check 2 "Password"  
wait  
if check 1 then output "My name^M"  
if check 2 then output "My password^M"
```

Set up the strings with a series of check commands, and then execute the wait command to go into terminal mode and begin looking for those strings. When one of the strings is detected, the wait command ends, and the script must determine which string was received with an if/then statement.

Strings defined with the check command are case sensitive, so you must include capital letters where necessary.

If a received string pattern matches more than one string specified by check commands, the lowest number is returned. For example, if string 1 is '>' and string 2 is Lobby>, and the string Lobby> is received, the script recognizes string 1 because it is a lower string number, even though both strings match the received data.

```
download "filename,type"
```

Downloads the specified file using the installed protocol. The one-

letter type specification is optional. If used, it must be either a `p`, `s`, or `u` for `prg`, `seq`, or `usr` file types, respectively. For batch protocols, the file name specified here is ignored, but be sure to at least put a pair of quotation marks to force the command to compile.

`hangup`

Hangs up the phone.

`minit`

Initializes the modem if no carrier is present.

`macro [function key number]`

Sends a function key string. This is equivalent to manually pressing one of the 16 function keys.

`offhook`

Takes the phone off the hook.

`output "string"`

Outputs a string to the modem. Control characters are entered as they are in macro keys, with a caret `^` preceding the corresponding letter. For example, a carriage return (CTRL-M) would be entered as `^M`.

`screen "string"`

Prints a text message on the screen.

`send "filename"`

Sends an ASCII file to the modem (equivalent to `Textfile to modem` on the `Disk` menu).

`uncheck [string number]`

Clears the string in the specified string number. For example, if the command `check 2 "Password"` were previously executed, `uncheck 2` gets rid of the definition for string #2.

`upload "filename"`

Uploads a file using the installed protocol.

`wait`

Goes into terminal mode and receives data (and prints it to the screen) while waiting for the strings specified with the `check` command.

When one of the strings is found, Novaterm stops waiting and continues with the script.

6.2.2 Dialing

dial "entry name"

Dials the entry with the given name. The autodialer keeps redialing until either a carrier is found or the redial count is exhausted (see redial).

dial# "phone number"

Dials the phone number specified.

dialn [location in phone list]

Dials the entry in the phone list corresponding to the specified number. The numbers don't show up on the autodial menu, but, for example, the command dialn 3 would dial the third entry from the top.

redial [number of tries]

Sets the number of times to redial a number before quitting. If a number is dialed this many times without finding a carrier, Novaterm aborts the script.

6.2.3 Communication parameters

baud [baud rate]

Changes the baud rate to the specified number. If the baud rate given is greater than the maximum baud rate allowed by the modem and serial drivers, the maximum baud rate is used instead.

param "setting"

Changes the current word length and parity setting. Valid arguments for the setting are:

```
param "8n1" ; 8 bits, no parity
param "7e1" ; 7 bits, even parity
param "7o1" ; 7 bits, odd parity
```

6.2.4 Buffer commands

buffer <command> <"filename">

The following commands are valid with BUFFER:

buffer close - Closes a capture file.
buffer kill "filename" - Deletes a file from the buffer.
buffer load "filename" - Loads a file from disk into the buffer.
buffer open "filename" - Opens a capture file.
buffer print "filename" - Outputs a buffer file to the printer.
buffer save "filename" - Saves a buffer file to disk.
buffer send "filename" - Sends a buffer file to the modem.

6.2.5 Installing drivers

For commands which install drivers, omit the prefix from the file name. (For example, the command `prot"Xmodem"` would work, but `prot"prt.Xmodem"` would not.)

asc "ASCII table" - Loads an ASCII translation table.
font "font file" - Loads a font.
font80 "font file" - Loads an 80-column font.
modem "modem type" - Loads a modem driver.
prot "protocol" - Loads a protocol.
term "terminal emulator"

Loads a terminal emulation. Note that this only loads the term file specified, *not* the asc and font80 files associated with it.

go40

Changes to 40-column mode, if the installed terminal emulator allows it. (With ANSI, VT102, or VT52, this will have no effect.)

go80

Changes to 80 column mode.

6.2.6 Disk operations

disk [device number]

Changes the current disk device to the specified device number. If the device is not present, the script halts with an error.

diskc "disk command"

Sends the specified disk command to the current disk device. The extra disk commands added by Novaterm are also valid here.

drive [drive number]

Changes the current drive number (not device number).

```
print "filename"
```

Sends a disk file to the printer. If the printer is not turned on or connected, the script aborts.

```
printer [device number]
```

Changes the printer device to the specified number. If the device is not present, the script aborts with an error.

```
secaddr [secondary address]
```

Changes the printer secondary address.

6.2.7 Script program control

```
end
```

Terminates the script program.

```
if <no> (condition) then (command)
```

Used to test various conditions. Any one of the conditions below may be substituted into the command, and if the condition is true, the command after then is executed. Any command may be put after then.

if carrier - Checks for a carrier.

if check [string#] - Checks to see if the specified string number was received. There must be a number after check, or the command is invalid.

if receive - Checks to see if data is being received.

if ring - Checks to see if the phone is ringing.

```
jump label
```

Jumps to the specified label in the program. Labels are non-command words which are placed before commands to identify a location in the script. For example:

```
loop check 1 "Username:"  
..  
..  
jump loop
```

Labels may contain any characters other than quotes, but a label may only be one word.

```
link "script program"
```

Loads and executes a new script program. Execution does not return to the current script.

6.2.8 Time commands

`gettime`

Attempts to read the current time from the real-time clock device (see 1.1.10, **Selecting a real-time clock device**). If the clock device selected is Manual, this command requests the user to input the current time. If your script program uses the `until` command to wait for a certain time of day, you should include this command at the start of your program to make sure the clock is set correctly.

`pause [number of seconds]`

Pauses the program for a specified number of seconds.

`until "24-hour time"`

Pauses execution of the script until the system clock reaches the specified time. The time must be in 24-hour form. For example:

```
until "22:35" ; waits until 10:35 PM
until "11:50:25" ; waits until 11:50 AM and 25 seconds
```

7. UTILITY MODULES

Novaterm provides a set of external programs to supplement the functionality of the software. This section describes the operation of each of these programs (with the exception of the Text Editor and BBS Mode, which are covered in Appendices A and B).

7.1 ASCII table editor

The ASCII table editor allows you to modify the mappings used to convert incoming data into readable text. Because the Commodore 64 assigns different ASCII values to characters than the rest of the world, these tables are necessary to convert incoming characters into values that are proper for the Commodore.

Each file actually contains two tables: a "transmit" table and a "receive" table. The transmit table converts your keystrokes into ASCII values that can be transmitted over the modem; the receive table converts incoming characters into ASCII values that can be displayed on your screen. There are 256 possible characters, and each one has a mapping in both the transmit and receive tables.

When you run the ASCII table editor, you are asked for the name of a translation table. After you specify the name of a table, the first 64 values in the transmit table is displayed, along with a list of commands at the top of the screen. The table is laid out such that the first number in each entry is the value before conversion, and the second number is the converted value.

The cursor keys move the highlight bar around the visible portion of the table. The following commands are also available:

Key: Modify the converted value of the current selection by pressing a keyboard character. The ASCII value of the key you press appears as the converted value.

Number: Modify the converted value of the current selection by typing in a number.

Page: Switch to the next 64 values in the current table.

Table: Toggle between the transmit table and the receive table.

Load: Load a new ASCII table.

Save: Save the current ASCII table.

7.2 Font editor

The font editor allows you to modify the appearance of characters as they are displayed on the screen. The font editor handles both 40-column and 80-column fonts. When you run the font editor, it initially displays the default 40-column font, which is copied from the Commodore 64's character ROM. You can switch between 40-column and 80-column mode by pressing the X key.

Loading a font and editing characters involves the same key commands in either mode. To load a font into memory, press the L key and supply the name of a font file.

Use the cursor keys to select a character to edit. The character number is displayed, and a blown-up version of the character is shown for editing. To edit the character, press RETURN. In edit mode, use the cursor keys to move the asterisk, and use the SPACE key to turn a pixel on or off. As you modify the character, you'll see the change take effect in the actual-sized character. Press RETURN again when you are finished editing. When you are finished editing the font, press S to save it.

The 40-column font editor has a number of extra commands that are useful for manipulating characters:

Reverse: Invert every pixel in the current character.

Flip: Do a vertical flip of the character.

Mirror: Do a horizontal flip of the character.

Rotate: Rotate the character 90 degrees clockwise.

Upspin: Shift the character up one pixel.

Downspin: Shift the character down one pixel.

Leftspin: Shift the character left one pixel.

Rightspin: Shift the character right one pixel.

Fill buffer: Copy the current character into the buffer space.

Write buffer: Copy the pattern in the buffer to the current character.

Copy chars to reverse: Copy the first half of the character set to the second half, reversing each character at the same time.

7.3 Configuration

The configuration program is started automatically if you try to run Novaterm without a configuration file. It can also be used at any time to reset the configuration to a new state. See section 1.2, **Starting up the program** for a description of the options in the configuration program.

APPENDIX A. TEXT EDITOR

Nova-Text is a utility for creating and editing textfiles (sequential files). Nova-Text provides a number of editing functions accessible with control key combinations. A list of editor commands is available on a single help screen from within the program by pressing `CTRL-H`.

Nova-Text uses as much contiguous internal memory as possible for editing textfiles. This means that you may edit textfiles up to about 38K in size (about 39,000 bytes, or 150 disk blocks).

A.1 Status line

Nova-Text uses a status line at the top of the screen to indicate various settings. From left to right, you'll see the following information:

1. Current line and column position of the cursor. `L` indicates the line number and `C` indicates the column number.
2. A flag indicating insert (`Ins`) or overwrite (`Ovr`) mode. In insert mode, any text you type is always inserted before the text under the cursor. In overwrite mode, text under the cursor is overwritten by your typing.
3. The current disk device and drive/partition number. If the current device is set to the buffer, the word `Buf` appears here.
4. The name of the file being edited, if there is one.
5. An asterisk next to the file name indicates the current file has been modified.

A.2 Loading a file

To load a file into the editor, press `CTRL-L`. Type in the name of the file.

A.3 Entering text

To enter text in the editor, simply start typing. To move the cursor around, use the arrow keys. Additional navigational commands are provided:

- `F1`: Page up - moves up a page.
- `F2`: Top of file - moves the cursor to line 1.
- `F3`: Page down - moves down a page.
- `F4`: Bottom of file - moves the cursor to the last line.
- `F5`: End of line - moves the cursor to the end of the current line.

F6: End of screen - moves the cursor to the last line on the screen.
F8: Tab - jump to the right 8 spaces.

A.4 Deleting lines

To delete the current line, press CTRL-Y.

A.5 Finding and replacing text

To search for a string, press CTRL-F. Type in the string to search for. The cursor immediately jumps to the first instance of the string after the current cursor position. To jump to the next instance of the string, press CTRL-V.

To search for and replace a string with another string, press CTRL-R. Type in the the string to search for. Then, type in the string to replace it with.

Nova-Text will ask if you want to replace all occurrences of the string. If you answer no, only the first string found will be replaced. Press CTRL-V again to replace the next occurrence of the string.

If you answer yes, all occurrences of the string are replaced, and Nova-Text tells you how many changes were made.

A.6 Changing margins

Nova-Text allows you to edit text up to 255 columns wide. Even though not all columns are visible at once on the 40-column screen, you can scroll the screen to the left and right by moving the cursor beyond the right edge of the screen.

To set the margins, press CTRL-M. Enter the column number for the left and right margins. When you have set the margins, all word wrapping will stay within those boundaries.

A.7 Formatting text

Nova-Text will automatically format text to the margins you specify while you type, as long as you have word wrap turned on. To toggle word wrap, press CTRL-W.

If you change the margins and want to reformat a previously typed paragraph, move the cursor to the first column of the first line of the paragraph, and press CTRL-A.

A.8 Moving, copying, and deleting text

Nova-Text allows you to mark off a section of text to move, copy, or delete. This involves four steps:

1. Move the cursor to the beginning of the text you want to mark, and press **CTRL-B**. Nova-Text tells you that you have marked the start of the block.
2. Move the cursor to the end of the text you want to mark, and press **CTRL-B** again. At this point, the text section is highlighted, and Nova-Text asks whether you want to move, copy, or delete the text.
3. Press **M** to move, **C** to copy, or **D** to delete. Pressing any other key will unmark the text and abort the operation.
4. If you decided to move or copy the text, move the cursor to the new position for the text and press **RETURN**. No other editing is possible until you have pressed **RETURN**. The status line shows the letters **Blk** before you press **RETURN** to indicate this special mode.

A.9 Saving a file

To save the file, press **CTRL-S**. Type in the file name. To use the previous file name, just press **RETURN**.

A.10 Compiling a script

To compile the text in the editor into a script file, press **RUN/STOP**. See section 6, **Scripts** for more information about this.

A.11 Other commands

The following commands are also available in the text editor:

CTRL-C: Send a disk command to the current device.

CTRL-E: Erase memory - Clear out the editor.

CTRL-H: Help screen - Bring up the help screen

CTRL-I: Input filter - Turn the input filter on or off. The input filter removes any non-ASCII characters while a file is loaded.

CTRL-K: Append file - Append a file to the file in the editor.

CTRL-N: Insert line - Insert a blank line before the current one.

CTRL-O: Bytes free - Display the number of bytes free.

CTRL-P: Print file - Print the file on the printer.

CTRL-T: Set tab size - Set the tab width (F8 is tab).

CTRL-X: Delete char - Delete the character under the cursor.

CTRL-SHIFT: Toggle character mode - In character mode, every key you typed is displayed in the editor, including control keys.

A.12 Other DOS commands

The following commands may be entered at the DOS prompt:

@ - Read error channel

\$ - Directory

- Change device and drive

If you specify the letter B as the device number, the editor will use the buffer as the current device.

APPENDIX B. BBS MODE

BBS mode is a small answering service of sorts that imitates a bulletin board system. It allows users to call in, view the directory, upload and download files, read and write textfiles, and leave messages to you. It has a chat mode, and it keeps a log of people who have called. Users may also upload hidden files, which may not be seen by other callers.

Unlike a real BBS, BBS mode has no public message base, no electronic mail (except feedback to the sysop), and no maintained list of users. The BBS mode is meant only as a simple front-end interface for users to call in and get files.

B.1 Configuration file

BBS mode reads your standard configuration file for Novaterm, and installs the modules specified there.

In addition, BBS mode uses its own configuration file (given the file name “\bbs config”), and this is loaded when the program is run, if it exists. The program also keeps a statistics file (called “\stats”) which records the last caller, number of messages, number of uploads and downloads, and number of callers.

B.2 Initial setup

BBS mode first tries to set the system clock using the module you have specified in your configuration for Novaterm (see 1.1.10, **Selecting a real-time clock device**). If you have selected the Manual module, BBS mode asks you for the current time.

You are then presented with the setup menu. Each option has the following function:

Sysop name: This is the name you use to log in as the sysop. When you log in as the sysop, you have special access to read feedback, execute disk commands, look at the caller log, and view and download hidden files. If you enter this name when you log in, you are prompted for sysop password, described below.

BBS name: This is the name displayed above the default user menu. It has no other real significance.

System password: If you enter a system password here, users are required to supply the password when they log in. If this is left blank, no password is requested at login.

Sysop password: It is highly recommended that you enter something here. If you try to log in under the sysop name, this sysop password is requested. Supplying the correct password gives a person sysop access, described above. As a double protection against invalid sysop access, the sysop name is never made known to users who call in, so someone would have to know both the sysop name and password to get sysop access.

Opening message: This is the file name of a textfile that is displayed after someone logs in. If left blank, the file is not searched for.

Main menu: This is the file name of a textfile that contains the main menu. If left blank, the default menu is displayed.

Logoff message: This is the file name of a textfile that is displayed just before someone logs out. If left blank, nothing is displayed.

System drive: This is the device and drive number of the disk that is used for displaying the above textfiles, selecting protocols, saving the caller log, and saving the statistics file.

Available drives: This brings up a menu which allows you to specify which drives may be accessed with the Change drive command. You must give names to the drives. If you are going to run more than one drive on the BBS, you may divide files into different categories using the names of the drives.

Save configuration: This saves all of the current settings on the menu.

Load configuration : This re-loads the settings that were last saved.

Start BBS: When you are finished with the setup menu, select this option to set up BBS mode to answer calls. Pressing F1 also selects this option.

B.3 Waiting for a call

BBS mode goes into loop waiting for the phone to ring. You may interrupt the loop by pressing F1 to abort the program, F7 to return to the setup menu, or F3 jump into local mode (allowing you to log in). In local mode, the phone is taken off the hook so that callers get a busy signal.

B.4 Logging in

During a login, BBS mode asks for a name. If you enter the sysop

name, you are asked for the sysop password, as described above. The name entered is used to designate an author for feedback messages and to keep track of people who have called. If a system password is in effect, it is requested after the name.

APPENDIX C. VENDOR INFORMATION

This section contains addresses and phone numbers of vendors for the various hardware products listed in this documentation.

C.1 CommPort, BBGRam, RAMDrive, BBRTC, AddressFixer

The CommPort cartridge contains a 6551 UART chip and a standard RS232 serial port, enabling speeds up to 38,400 bps.

The BBGRam cartridge is a memory expansion device available in sizes of 512K, 1 MB, or 2 MB of RAM.

The RAMDrive is an intelligent, transparent memory expansion device. It is available in sizes up to 8 MB.

The BBRTC is a small, battery-backed real-time clock chip that plugs into either joystick port.

The AddressFixer is a simple in-line device that remaps the I/O address of a cartridge to a different location, and restricts the I/O register space of a device to 16 bytes (rather than allowing the device to mirror its registers over 256 bytes, as most devices do). The right combination of AddressFixers can make otherwise incompatible cartridge port devices work together by eliminating address conflicts. It resides in-line between a cartridge and a cartridge port.

The above hardware is sold by Performance Peripherals, Inc.

Performance Peripherals, Inc.
5 Upper Loudon Rd.
Loudonville, NY 12211 U.S.A.
(800) EASY-WEB
E-mail: fiset@global1.net

C.2 SwiftLink, RAMLink

The SwiftLink cartridge contains a 6551 UART chip and a standard RS232 serial port, enabling speeds up to 38,400 bps.

The RAMLink is an intelligent memory expansion device that uses SIMM chips. It is expandable up to 16 MB.

The SwiftLink and RAMLink are both sold by Creative Micro Designs, Inc.

Creative Micro Designs, Inc.
P.O. Box 646

East Longmeadow, MA 01028 U.S.A.
(413) 525-0023
E-mail: doug.cotton@the-spa.com

C.3 HART cartridge

The HART cartridge uses an 8250 UART chip connected to a standard RS232 serial port, achieving speeds up to 57,600 bps. The HART cartridge is sold by Hatronics.

Hatronics
c/o Mark Hatten
145 Lincoln Street
Montclair, NJ 07042 U.S.A.
E-mail: markhatten@att.com

C.4 Enhanced RS232 interface (UP9600)

Daniel Dallmann has designed a simple modification to the standard RS232 interface (which plugs into the user port) that allows the Commodore 64 to operate at up to 9600 bps. Information on how to make this modification is available on the Internet through the World Wide Web, at the following site:

<http://rpool1.rus.uni-stuttgart.de/~etk10217/proj.html>

E-mail: Daniel.Dallmann@studbox.uni-stuttgart.de