

SOFTWARE MANUAL

# Operation



# VersiCOMM<sup>Plus</sup>

•Valley Programming Service•

THE INFORMATION CONTAINED IN THIS MANUAL IS BELIEVED TO BE  
ACCURATE AND RELIABLE. HOWEVER, NO RESPONSIBILITY FOR THE  
ACCURACY, COMPLETENESS OR USE OF THIS INFORMATION IS  
ASSUMED BY VALLEY PROGRAMMING SERVICE, INC.

## VersiCOMM OPERATION MANUAL

© 1995 VALLEY PROGRAMMING SERVICE, INC.

MANUAL REVISION 3.3 - COMM REVISION 3.1 - PDF

May 1, 1995

VersiCOMM and VersiCOMM-Plus are trademarks of Valley Programming Service, Inc., Canoga Park, CA 91303.

AMOS, AlphaBASIC and Alpha Micro are trademarks of Alpha Microsystems, Santa Ana, CA 92799.

FasTalk is a trademark of Universal Data Systems, Inc., Huntsville, AL 35805

Compuserve is a registered trademark of Compuserve Inc., an H&R Block Company.

Valley Programming Service, Inc.  
21822 Sherman Way Suite #202  
Canoga Park, CA 91303  
(818)-992-5134

# Contents

---

## Contents 3

## Overview 11

Operation	13
Conversational Mode	13
Command Mode	14
Special Features	14
Internet Access Support	14
FastQ™	15
FastTrans™ File Compression	15
VersiCOMMander	15
Record and Playback	15
File List Processing	15
Modem Polling Files	15
Automatic Download Initiation	15
COMM LOG File	16
INI.TIW File	16
INI.TLK File	16
INIARG.TLK File	16
USING Files	17
The AMOS Command	17
Translation and Filtering	17
Extended Function Interface Drivers	18
HELP System	18
VT100/102 Terminal Emulation	18
Generic Terminal Emulation	18
MCI Mail and EasyLink QuikLetters	18
Input Flow Control	18
Soft Parity	19
Preserve Screen	19
Status Line	19
Idleout Autoexit	19
Command Editing and Recall	19

## Operation 21

Command Format	21
System Configuration - The INI.TIW File	23
VersiCOMM Initialization - The INI.TLK File	24
VersiCOMM Initialization - The INIARG.TLK File	24
Conversational Mode	25
VersiCOMM Status Line	25
Saving Displayed Information	26
Recording Displayed Information	26
Generating a 'BREAK'	27
Changing to VersiCOMM Command Mode	27
Direct Exit from Conversational Mode	27
Preserving the Conversational Mode Screen	27
Auto-Timeout From VersiCOMM	27
Command Mode	28
The Command Mode Status Display	28
Operational Switches	28
Port Selected	29
Port Configuration	30
SEND Parameters	30
Control Keys	31
Entering VersiCOMM Commands	31
VersiCOMMander File Manager	32
The Main Window	33
Selecting Files	33
The Status Window	34
File Submenu	34
Specify	34
View	34
Hash	35
Copy	35
Rename	35
Erase	35
Set All	35
Clear All	35
Enter Command	35
Exit	36
Send Submenu	36
Enter Command	36
FasTrans	36
Zmodem	36
Ymodem	36
Xmodem	36
Xmodem 1K	36
B-Plus	36
Receive Submenu	36
Enter Command	36
FasTrans	37
Zmodem	37
Ymodem	37
Xmodem	37
Xmodem 1K	37
B-Plus	37

Settings Submenu	37
Query	ON 37
Delete	OFF 37
IND File	OFF 37
Erase Ctrl	ON 38
Destination	38
System Submenu	38
System Status	38
Network Status	38
System Info	38
Terminal Defs	38
Ersatz Names	38
Translating & Filtering Characters	38
Filtering Nulls	39
The AMOS Command	39
Command Files, Task Manager & Shell	39
The COMMCENTRE	40
Dialing Methods	40
Dialing in Conversational Mode	40
VersiCOMM DIAL Command	41
VersiCOMM Script File	41
The DIALER	41
MLON and MLOFF	41
MATCH	41
Operation	41
Switches	42
Errors	43
TRMSTS	43
CHGTDV	44
Operation	44
Errors	45
Filter	45

## **The DIALER 47**

Overview	47
DIALER Data Files	47
DIALR5.BAS Source File	47
Function Key Operation	47
DIALER Subroutines	48
The Dialer Screen	48
Selection Controls	48
Function Key Actions	49
Entry Screen	50
Entry Screen Fields	50
Field Editing	53
Function Key Actions	53
Modem Setup Screen	54
Modem Setup Screen Fields	54
Field Editing	56
Function Key Actions	56
Dialer Example - Accessing CIS	57
Dialer Utilities	59

## **'USING' Files 61**

Introduction	61
Using File Format	62
The WAIT Command	64
Using File Operation - The WASIT Command	65
Using File Operation - The CTRLC Command	65
Using File Operation - The DIALOG Command	66
The LISTEN Command	67
USING File Termination	68
OFF Files	68
HANGUP File	68

## **Sending and Receiving Files 71**

Transmitting Files	71
SEND	72
Speed Control	72
End-of-Line Delay	73
Flow Control	73
Linefeed Control	73
Translation and Filtering	73
SENDEPX	73
SENDL	74
SENDTLK	74
TRANS	75
Operation	76
File List Processing	78
Option Switches	79
Troubleshooting	83
Error Messages	84
SENDX AND SX	87
SENDX Operation	88
SX Operation	88
Switches	89
Errors	89
SENDZ	89
Operation	90
Switches	90
Errors	91
Receiving Files	92
SAVE	92
RECEIVE	92
RETRV	93
Operation	94
Switches	97
Troubleshooting	100
Errors	101
RECX AND RX	104
RECX Operation	104
RX Operation	105
Switches	105
Errors	105
RECZ	106
Operation	107
Switches	107
Errors	108

KERMIT 109

## **BASIC Interface 111**

- Required Map Statements 111
- GCMARG and PCMARG XCALLS 112
- TREAD and TWRITE XCALLS 112
  - XCALL TWRITE 113
    - Examples of TWRITE Calls 113
  - XCALL TREAD 113
    - Examples of the TREAD Call 116
- XCALL TWAIT 116
  - Examples of the TWAIT Call 117
- GETPIN XCALL 118
- SETPIN XCALL 119
- ARGn and FLGn Values 120

## **Terminal Emulation 123**

- Virtual Terminal Emulation 123
- Specifying A Terminal Emulation 124
- VT100 Emulation 124
- Lexis Emulation 125
- Generic Terminal Emulation 126
- Additional Emulations 126
- Saving Displayed Text to Disk 126
- Recording Displayed Text and Playing it Back 126
  - Player Controls 127
  - Player Switches 127
    - Recording and Playing Back an Internet Session 127
    - Recording and Playing Back an Alpha Micro Session 128
- Keyboard Translation Files 128
- The VPSTRN Command 129

## **Commands 131**

## **VersiCOMM on the Internet 167**

- Scope of This Chapter 167
- What Terminal Should I Use? 167
- Establishing an Account 168
- Setting Up the Script 168
  - Creating the Files Manually 168
  - Using the COMMCENTRE 168
- Accessing Netcom 169
  - The Netcom System 169
  - The login Shell 169
  - Logging Off 169
- Getting Help 169
  - man 169
  - faq 170
  - Other Sources of Information 171
    - Unix System 171
    - World Wide Web 171
    - Gopher 172

Usenet	172
Command Recall	172
Command Aliasing	172
Editing a File	173
Introducing pico	173
Additional Information	175
File Management	175
ls	175
Additional Information	175
utree	175
Downloading Files Using utree	177
Additional Information	177
Browsing the World Wide Web	177
Starting lynx	178
I'm Waiting...and Other Hints	179
lynx Commands	180
Defining Your Home Page	180
Yahoo and Other Search Utilities	180
Additional Information	181
Browsing the Net Using gopher	182
Terminal Notes	182
You Have to Start Somewhere	182
Keyboard Controls	183
Navigation	183
Bookmarks	183
Other Commands	183
Searching Using Veronica	184
A gopher Sampler	185
How Is Your Car Running?	185
Is It Going to Rain Tomorrow?	185
Additional Information	186
Electronic Mail	186
First Things First...	187
Getting Help	187
Send a Test Message...	187
Read a Test Message...	188
Reply to the Test Message...	188
Saving the Test Message After Reading	188
Additional Information	188
Working with Newsgroups	188
Before You Start (Don't Skip This Step)	189
Getting Help	189
Your First Newsgroup	189
Finding Newsgroups to Read	190
Using rn to Locate Newsgroups	190
Using lynx to Locate Newsgroups	191
Posting a Reply to a Newsgroup Article	191
Additional Information	192
Terminal Access to Net Sites	192
Starting a telnet Session	192
Telnet Commands	193
An telnet Session to the CDEC	193
Additional Information	194
File Transfers	194



Known, But Anonymous	194
Starting a ftp Session	195
ftp Commands	195
Kodak PhotoCD Info via ftp	195
Additional Information	198
Finger	198
Additional Information	200

## **Examples and Experiments 201**

How To Talk To Your Modem	201
If You Cannot Talk To Your Modem	202
How To Dial And Connect To Another Computer	202
With a "Smart" Modem	202
With a Manual Modem	203
After Establishing A Connection	203
If No Connection is Established	203
How To Talk To Your Own Computer	204
Saving Displayed Information	204
Naming A Save File	205
Translating & Filtering Characters	205
How To Write A Using File	206
How To Send An MCI Letter	210
How To Send Batch EasyLink Messages	211
Accessing The AMUS Network Computer	212
Transferring Files To Or From AMUS	212
How To Use DJNS To Get Quotes And News	214

## **Errors 217**

## **Index**



## Overview

---

VersiCOMM-Plus is a versatile, general purpose communications program. It is not limited to performing just one communication task, such as sending Telexes or communicating with Compuserve, but can be used to efficiently perform a broad range of communications services. VersiCOMM is compatible with virtually all popular asynchronous modems.

In addition to COMM, the communications executive, the "VersiCOMM-Plus" package includes the following software:

- The error correcting "Alpha to Alpha" file transfer programs TRANS and RETRV. These programs, working in conjunction with their remote modules RZVA and TNZA, will transfer any type of AMOS file with *FastQ* input processing, CRC16 error detection, Adaptive or *FasTrans* Data Compression, packet sizing and data rate. Many advanced features are supported, such as the capability to transfer a file only if it's hash total is not the same as the file on the receiving system, or only if the version is later than the version of the file on the receiving system, and more. Both TRANS and RETRV will accept standard AMOS wildcard file specifications.
- The VersiCOMMander "point and select" Command mode file manager.
- The DIALER, a comprehensive dialing database and modem control file generator, including the AlphaBASIC source program DIALR5.BAS. Using the DIALER, the user can dial, connect, log on to, and off of most computers and services without having to write script files. VUE-like commands enable the user to scroll by line or page through the directory in order to select the number to dial. A search/indexing feature gives the user the ability to locate a particular entry without having to scroll through the entire file. A sort and expand program, EXPDLR.BAS, has been added.
- The "VPS COMMCENTRE". This is an AlphaMENU controlled communications center that illustrates how VersiCOMM can be used to provide "push-button" access to several remote services or even to provide "push-button" retrieval of mail, news, stock data and other information from remote services without operator intervention. The COMMCENTRE includes selections to automatically setup and access most popular communication and information services including the Internet via Netcom Online Communications.

- XMODEM file transfer software. With this software files can be transferred between an Alpha Micro and a PC that has a communications package using the very popular XMODEM error correcting protocol. Most PC communication packages and online computer services can use the XMODEM protocol. Either the Alpha user or the PC user may originate the transmission or receipt of a file. Both "checksum" and "CRC16" error detection are supported. XMODEM 1K packets are also supported.
- ZMODEM file transfer software. ZMODEM is a robust file transfer protocol originally developed to alleviate throughput problems experienced by packet-switched network customers using the XMODEM and KERMIT file transfer protocols. Unlike XMODEM, it does not pad the last block of a transferred file. Much faster than KERMIT, ZMODEM uses a streaming protocol with very low overhead, enhanced *FastQ* input processing, and a 32-bit CRC checksum. In addition, multiple files may be downloaded in a single transfer with automatic download initiation, and should the transfer be interrupted you can dial back into the system and resume the transfer at the point at which it was interrupted. The ZMODEM software also includes YMODEM batch file transfer support. YMODEM offers an alternative protocol for transferring multiple files to or from a system that does not support ZMODEM.
- Compuserve B Plus file transfer software. This software enables VersiCOMM-Plus to transfer files to, or from, the Compuserve Information Service using the B Plus protocol. Network induced delays can severely impede the speed of simple ACK/NAK file transfer protocols such as XMODEM. The B Plus protocol supports deferred acknowledgment of transferred packets, and while not as fast as a streaming protocol such as ZMODEM, is *much* faster than XMODEM in the Compuserve network environment. Automatic download and upload initiation is supported, as well as the resumption of an interrupted download.
- Subroutines that enable an AlphaBASIC program to: exchange data with a VersiCOMM script file; wait for specified strings to be received at the modem port; read data from, or write data to, the modem port.
- The baud-matching modem security program MATCH, which automatically adjusts the baud rate of a port to the speed of the terminal using the port, then prompts the user for an optional
- The utility CHGTDV. When placed on a remote system, CHGTDV.LIT can be used to switch terminal drivers "on the fly". The program will display drivers that are available in the Terminal Definition Chain or in System Memory. It will prevent changing to a driver that requires more impure area memory than is available.
- The utility TRMSTS. This program displays the state of all the status bits for a specified terminal, and enables the user to clear bits and conditions as required.
- VT100/102, VT52, VT220, LEXIS and Generic terminal emulation software, including VPSTRN, a program that enables the user to directly modify an emulation's keyboard translation file.
- SHWVT1, SHWV52, SHWLEX and SHWGEN. File "players" that play back a recorded remote session in simulated real time.



- The X COMMAND wildcarding program. Enables using commands such as COMPIL and TYPE as wildcard file commands.
- The MLON and MLOFF modem locking and unlocking commands. These commands are executed by a remote user to prevent a local user from using a modem port while the remote user is using the modem port.




## Operation

VersiCOMM operates in a manner similar to the AMOS program AlphaVUE. There are two modes of operation, Conversational and Command. The user switches back and forth (toggles) between these modes by keying a user modifiable character initially defined as <Control-^>. This character is generated by the <HOME> key on many terminals, so in the same way you use the <ESCAPE> key in VUE, you use the <HOME> key in VersiCOMM. *In case of program conflicts or user preferences, the various keyboard characters that control the operation of VersiCOMM may be redefined by the user in a special initialization file called an INI.TLK file.*

### Conversational Mode

Conversational mode can be thought of as being similar to the text entry mode of VUE, except that as text is typed in by the user, it is redirected and output via a communications port and modem to a remote computer. Any responses from the remote computer sent to the communications port are relayed and displayed on the user terminal. In Conversational mode you are "talking to" the remote computer and in effect, commands may be typed into the remote computer as if it were in the next room.

Most characters you type while in Conversational mode are sent along to the remote computer. There are a few special characters that are used for controlling the operation of VersiCOMM while it is in Conversational mode. As just discussed, typing a <Control-^> will switch VersiCOMM to Command mode. Other special characters include , , and <~>.

As information is being displayed on the terminal the user may choose to save the text in a disk file by keying a . This SAVE switch may then be repeatedly turned off and on by pressing , with text being saved at the user's discretion. A 'BREAK' condition may be generated by keying a . The user may exit VersiCOMM directly from Conversational mode (without having to toggle to Command mode) by keying the <~> character.

If the file "trmdef.CLG" exists in the user's account, VersiCOMM will append usage information to the file each time the user exits from the program. (Note that "trmdef" is the "terminal definition" of the modem, usually "MODEM1".)

## Command Mode

When you switch to command mode you will see a status display similar to the following:

```

VersiCOMM 2.1(385)
-----
On For Time Save Echo AutoLF TLocal TRemote Strip JWait
0 min 10:58 AM OFF OFF OFF OFF OFF ON ON
-----
Terminal Driver Interface Port Baud Rate Parity Size Stop DTR
TRM1 PSEUDO AM1000 0 19200 NONE 0 1 ON
-----
Linefeed Handshake EOLChr EOLTime Emulate BreakChr ChgMode DirSave DirExit
ON OFF LF 0.00 TTY ^@ ^M ^A
-----
Current Login
DSK1:[77.12]
>
-----
COMMAND | | 19200 | | | | SAVE OF

```

This display specifies a number of important parameters and switch settings. It is described in detail in the OPERATION chapter of this manual. As in VUE, the ">" character is a prompt for VersiCOMM commands.

Command mode is used to dial the modem, send and receive files, configure the modem port, set operational parameters, redefine program control keys, and to perform many other functions. From the command mode prompt you can run BASIC programs or execute AMOS LIT files, or get help information on over one hundred VersiCOMM-Plus commands. Status information such as the currently selected modem port, the port configuration, the time of day and the current login account are displayed while in command mode.

## Special Features

### Internet Access Support

VersiCOMM includes scripts and other enhancements to support modem access to the Internet via Netcom Online Communications. Using your Alpha Micro, and a terminal such as an AM65A, you can browse the World Wide Web, read and post articles to your favorite newsgroups, send and receive email, telnet to other sites, transfer files using ftp, do gopher searches using Veronica and more!

An entire chapter of this manual is dedicated to "VersiCOMM on the Internet." It describes the use of these applications:

lynx	World Wide Web browser.
elm	Electronic mail manager.
rn	Read and post articles on newsgroups.
gopher	Menu system for accessing net info and data.
telnet	Terminal access to computers on the net.
ftp	File transfers to and from net sites.
finger	Obtain prepared remarks from net sites.
pico	Easy to use file editor.
utree	OS/Exec like file manager.

VersiCOMM 3.1 has been extensively tested for compatibility with these applications. Just because you're using a character oriented terminal doesn't mean you can't go on the Information Superhighway!

### **FastQ™**

*FastQ*, an enhanced input processing technology, enables VersiCOMM-Plus to transfer files to small or heavily burdened systems with *dramatically* faster throughput and fewer retries. Originally incorporated into VersiCOMM-Plus's proprietary Alpha to Alpha file transfer software, the *FastQ* technology has now also been added to our ZMODEM software.

### **FastTrans™ File Compression**

Now, in addition to the very effective Adaptive compression technology already found in VersiCOMM-Plus's TRANS and RETRV programs, we have added the compression technology of the Z/Archive System, and achieved even higher compression ratios.

*FasTrans* gets the most out of your dialup and fixed serial data links by transferring more data in less time. Much more data in much less time! In one series of tests the AMOS 2.2C file IDV681.M68 was transferred from an AM3000M to an AM3000VME over a 9600 bps link at a transfer rate in excess of 34500 bps! Used in combination with *FastQ*, our super efficient input system, *FasTrans* speeds your AMOS data to its destination in record time.

### **VersiCOMMander**

By entering the command "SELECT" in VersiCOMM-Plus command mode you can display a scrollable window of files and select the files you want to transfer, view, copy, rename, hash or erase. Pull down menus makes it easy!

### **Record and Playback**

VersiCOMM-Plus includes tools to record and playback terminal sessions in simulated real-time. This feature can be used to create software demonstrations that can be played on other computers, to record online sessions for later review, or just to share your latest Internet find with a friend.

Players are included for our VT100/102, V52, LEXis, and GENeric terminal emulations.

### **File List Processing**

The VersiCOMM-Plus program TRANS can process a list of files, such as a VERIFY directory, and transmit each file specified in the list.

### **Modem Polling Files**

Modem polling files define the sequence of modems that are to be used by a particular application. These files have a default extension of "MPF" and reside in DSK0:[7,0]. They are specified in the COMM command line in the same manner as a modem.

### **Automatic Download Initiation**

VersiCOMM will now automatically detect the start of a ZMODEM download and initiate the ZMODEM receive program RECZ. The ZAUTO command is used to turn automatic ZMODEM downloading ON or OFF. The default is ON.

VersiCOMM can also automatically detect the start of a Compuserve B Plus protocol file transfer. The BAUTO command is used to turn automatic file transfer initiation ON or OFF. The default is OFF.

## COMM LOG File

If the file "trmdef.CLG" is found in the user account, VersiCOMM will add a line of usage information to the file. Since the modem "trmdef" is usually "MODEM1", the name of the LOG file is usually "MODEM1.CLG". In this case you would start the file by entering the command "MAKE MODEM1.CLG" at AMOS command level, or use VUE to create the file "MODEM1.CLG". Log file entries include the date, time the communication session ended, the total time for the session, the AMOS job name, the AMOS terminal name used for the communication session, and the name of the TLK file used to initiate the session. For example,

31-Aug-86	12:45:32	5 Minutes	JOB1	MODEM1	LEXIS
31-Aug-86	16:20:00	2 Minutes	JOB1	MODEM1	MCIMBX
31-Aug-86	19:33:43	10 Minutes	JOB1	MODEM1	FYI
31-Aug-86	20:04:28	12 Minutes	JOB1	MODEM1	FYI
01-Sep-86	08:04:12	1 Minutes	JOB1	MODEM1	MCI
01-Sep-86	08:29:15	2 Minutes	%TSKAA	MODEM1	MCIMBX
01-Sep-86	09:46:09	1 Minutes	%TSKAA	MODEM1	ELMBX
01-Sep-86	09:51:32	2 Minutes	%TSKAA	MODEM1	FYINWS
01-Sep-86	09:55:12	2 Minutes	%TSKAA	MODEM1	FYISPT
02-Sep-86	13:44:32	1 Minutes	JOB1	MODEM1	
02-Sep-86	13:46:27	2 Minutes	JOB1	MODEM1	AMINFO
04-Sep-86	16:06:18	1 Minutes	JOB1	MODEM1	MCI
04-Sep-86	16:08:41	1 Minutes	JOB1	MODEM1	NWSNET

## INI.TIW File

If a job attached to a modem is "waiting for input" or is "sleeping" COMM will check a list of programs maintained in the file DSK0:INI.TIW[7,0] to see if its OK to detach the modem from that job while it is running that program and waiting. For example, you would probably let COMM use MODEM2 if the job it was attached to was running a LOGON program and waiting for input, but you might not want to let it use MODEM2 if the program was ORDER.

## INI.TLK File

Whenever VersiCOMM is executed it checks to see if there is an 'INI.TLK' file in the user account or in DSK0:[7,0]. This is a text file that may contain VersiCOMM commands which redefine the various characters used to control VersiCOMM. For example you could change the default definition of the "change mode" character from <Control-^> to a backslash, <\>, by creating an INI.TLK file containing the command "\$CHGMODE 92". See the OPERATION chapter of this manual for more information on the use of an INI.TLK file.

## INIARG.TLK File

The INIARG.TLK file is similar in operation to the INI.TLK file, but it has a different purpose. The INI.TLK file is a static initialization file that remains (more or less) the same each time VersiCOMM is executed. The INIARG.TLK file is a dynamic initialization file that is used to set the value of the VersiCOMM arguments ARG0-ARG15, and flags FLG0-FLG15, and to set or configure other VersiCOMM switches and options. The INIARG.TLK is session-oriented, i.e. it is intended for use during a particular session of VersiCOMM. In fact, VersiCOMM erases the INIARG.TLK file after it is processed.



Whenever VersiCOMM is executed it checks to see if there is an 'INIARG.TLK' file in the user account. If there is, VersiCOMM executes it, then erases it.

Usually the INIARG.TLK file is created on the fly in another program which then chains to VersiCOMM.

## USING Files

In communications it is especially inconvenient to have to manually configure a program for communicating with a particular host (other computer system) and then to lookup and enter passwords, logon sequences, and other commands every time you want to access that system. It would be much easier to enter a single command, or make a menu selection and let the computer do the "remembering" and the work of accessing other systems. VersiCOMM's "USING" files provide this capability. A USING file can be thought of as being an operator of VersiCOMM. It can operate VersiCOMM much the same as if a person was at the keyboard typing in commands and other information.

USING files are just VUE files that contain VersiCOMM commands and text that would normally be typed at the keyboard. When a USING file is processed the VersiCOMM commands are executed and "keyboard" text is transmitted to the host computer. Used with a smart modem, a USING file can configure VersiCOMM to operate in particular way, then dial your modem, wait for a connect and password prompt, enter the appropriate password and logon sequence, wait for an acknowledgment and then when that's all done, turn operation of VersiCOMM back to you.

USING files can do automatically much of keyboard entry that an operator would do manually. Like VUE files, the number of USING files you create is only limited by the available disk space. A particular USING file is selected by specifying it in the VersiCOMM command line. For instance, COMM USING MCI could access MCI Mail, while COMM USING ELINK could access EasyLink.

In addition to accessing a service, USING files can be used to automatically send a service a frequently used set of commands.

## The AMOS Command

In addition to VersiCOMM commands, *most AMOS commands, such as DIR, SYSTAT, TYPE, and many others, may be executed from VersiCOMM's command mode* without having to exit to the monitor, by using VersiCOMM's "AMOS" command. The AMOS command also provides a convenient way to extend VersiCOMM to include special features or capabilities required by a particular user.

## Translation and Filtering

VersiCOMM has the capability to translate characters that originate at the "local" computer, i.e. the characters you type on a keyboard or send in a text file, before sending them to the "remote" computer, and it also can translate characters received from the "remote" computer before using them "locally". This is accomplished by building translation tables for remote and local characters. An entry in the local or remote translation tables is made using the LOCAL or REMOTE commands. Translation and filtering can be enabled or disabled without affecting existing tables using the TLOCAL and TREMOTE commands. This facility can be very useful for editing unwanted nulls and other control characters from data received from information and communications services. It can also be used to customize the operation of your terminal for a particular service without having to write a special driver.

## Extended Function Interface Drivers

VersiCOMM release 2.2 and later supports several commands which have been implemented as function calls to specially modified interface drivers. Prior to VersiCOMM release 2.7B these extended function interface drivers were designated "AX" or "AY" drivers, e.g. "AX1000.IDV." The standard "AM" drivers included with AMOS 1.3D/1.0D and later versions of the operating system now include support for the extended functions. VersiCOMM 2.7A and later versions of VersiCOMM are compatible with these new format drivers.

VersiCOMM commands which use the extended functions of the driver are PARITY, SIZE, STOP, DTR, DTRLOW, and BREAK. The SPARITY command does not require an extended function driver to change from the standard "eight data bits, no parity and one stop bit", to seven data bits and even or odd parity with one stop bit.

## HELP System

The VersiCOMM HELP system can display a list of VersiCOMM commands or provide information on any individual command. Enter HELP at the VersiCOMM command prompt.

## VT100/102 Terminal Emulation

This is a comprehensive emulation of Digital Equipment Corporation's VT100 terminal in 80 column, ANSI mode. The emulation supports use of the 14 character keypad in Application or Numeric mode, the cursor keys in ANSI or Application mode, Scrolling Regions, PF Keys, New Line and Autowrap modes. The reduced intensity attribute is supported; and if the terminal running the emulation supports "non-space" attributes, reverse, underline and blinking attributes are also supported.

## Generic Terminal Emulation

When used in combination with the Generic terminal driver on a remote Alpha Micro, this emulation provides terminal independent support for terminal specific functions such as cursor addressing, reduced intensity, reverse video, line drawing, etc.

## MCI Mail and EasyLink QuikLetters

In addition to the other mail functions found in the COMMCENTRE, users that subscribe to MCI Mail and EasyLink may now quickly and easily create and send any of several different types of messages by using MCI Mail and EasyLink QuikLetters. Not only electronic messages such as telexes and Email, but also fax messages and paper mail such as laser printed overnight letters may be written and mailed. (The MCI Mail QuikLetter uses a certified implementation of MCI Mail Link, a special protocol used to communicate with MCI Mail.)

## Input Flow Control

Input flow control can be used with 2681 and 68681 based interfaces to prevent the loss of characters due to high baud rates or an overburdened processor. Input flow control can be essential when linking one computer to another or when linking a computer to a device such as a protocol converter. It requires that the transmitting device be controllable via a RS-232 input and that the interface cable connect pin 5 of the Alpha Micro to the controlling pin of the device, usually pin 4.

In addition to hardware flow control, "software" input flow control is now supported by VersiCOMM using XON/XOFF inband control characters.

## Soft Parity

The most common non-Alpha Micro compatible parity requirement is for "seven data bits, even parity, and one stop bit." Using VersiCOMM's SPARITY command the communication port's parity can be changed to this configuration without using an extended function interface driver.

## Preserve Screen

If your terminal is an AM65A, or other terminal that supports the Save Screen Area function, you may use the PRESERVE command to preserve information that is displayed in conversational mode when switching back and forth from conversational to command mode.

## Status Line

If your terminal supports an addressable status line, user information such as the VersiCOMM mode, baud rate, connect rate, wait status, save status, etc., is displayed on the bottom status line.

## Idleout Autoexit

VersiCOMM can be configured so that if it remains inactive for a specified amount of time, either in Conversational or Command mode, it will automatically exit, just as if the user had exited the program.

## Command Editing and Recall

Commands entered at the VersiCOMM Command mode prompt can be recalled and edited using a full complement of editing keys.



## Operation

---

### Command Format

The generalized format of the VersiCOMM command line is

```
.COMM {terminal/MPF-name}{USING filespec}{WITH param0 param1...param15}
```

To run COMM type COMM, followed by the name of the terminal that is connected to your modem or direct link to another computer:

```
.COMM TRM4
```

If the name of this terminal is MODEM1 it is not necessary to specify the terminal name:

```
.COMM
```

A modem polling file name may be substituted for the terminal name. Modem polling files define the sequence of modems that are to be used by a particular application. These files have a default extension of "MPF" and reside in DSK0:[7,0]. For example, suppose the file "SEQ1.MPF" contains the following:

```
MODEM1  
FAX2  
MISC1
```

When the command,

```
COMM SEQ1 USING CLIENT1
```

is executed, COMM will first try to use MODEM1, then FAX2, etc.


If a modem polling file has the same name as a modem, the modem polling file will be used. For example, consider the following command:

```
COMM USING CLIENT1
```

Since no modem is specified, COMM, by default, uses MODEM1. If the file "DSK0:MODEM1.MPF[7,0]" does *not* exist then VersiCOMM will try to use the default

modem, MODEM1. However, if "DSK0:MODEM1.MPF[7,0]" *does* exist, then COMM will use the sequence of ports defined in that file. This provides a means for easily redefining the default modem. For example, if "DSK0:MODEM1.MPF[7,0]" lists MODEM9, then in effect MODEM9 becomes the default modem.

If a script file is to be processed it should be specified as in the following command line.

```
.COMM USING ELINK 
```


If the extension is omitted from the USING filespec, the extension TLK is assumed.

VersiCOMM uses the following sequence in locating a .TLK file specified as an argument to a USING parameter:

1. Account specified in USING file specification.
2. Account specified in an INI.TLK file MODEM command.
3. User account.
4. DSK0:[7,121] if not found any other way.

If required, up to three 30 character strings, and thirteen 60 character strings can be passed to a VersiCOMM USING file via the VersiCOMM command line. The character strings param0, param1, param2, etc. will replace any occurrence of "@0", "@1", "@2", etc. in a USING file command line. The @3 token is reserved for use by the DIAL command. Tokens greater than @9 are referred to as @A, @B, @C, @D, @E and @F.

In the following command line,

```
.COMM USING MCI WITH IBM.STK 
```

the string "IBM.STK" would be stored by VersiCOMM and could be referenced by the token @0. This would result in the USING file command "\$SAVE @0" being interpreted as "\$SAVE IBM.STK".

A DO file can be used to pass a DO command line argument to a USING file. The DO file QUOTE.DO might take the string "IBM", for example, and pass it to a USING file using the command "QUOTE IBM". The content of the DO file might be:

```
:R  
COMM USING DJNS WITH $0 
```

And then a line of the USING file DJNS.TLK might have:

```
$SENDTLK @0
```

The result would be that the command \$SENDTLK IBM would be processed.

The @n tokens can only be specified as part of a VersiCOMM command. If you need to write "token" data to a port as part of a USING file procedure use the WRITE or WRITECR commands.

## System Configuration - The INI.TIW File

When VersiCOMM executes it automatically configures the system for communications. As part of this process VersiCOMM must prevent the use of a modem port that is already in use for either outgoing or incoming communications. It also must be able to determine when it is OK to "borrow" a modem from another job. VersiCOMM makes these decisions based upon the following rules. ("Attached" means the job and terminal are linked to each other, and "linked" means the terminal is linked to the job but not vice versa.)

A modem (TRMDEF) is considered to be "busy" in the following situations:

1. Modem is marked "in use" for communications.
2. Modem is marked as having "output in progress."
3. Modem is linked to another job.
4. Modem is attached to another job, and the program running is not waiting for input.
5. Modem is attached to another job, the program running is waiting for input and it is NOT listed in the file DSK0:INI.TIW[7,0].

A modem (TRMDEF) is considered to be "free" in the following situations:

1. Modem is not linked or attached to another job.
2. Modem is attached to another job and no program is being run.
3. Modem is attached to another job, the program running is waiting for input and it is listed in the file DSK0:INI.TIW[7,0].

If the modem port specified for communications is MODEMn and VersiCOMM finds that the port is busy, it will automatically look to see if MODEMn+1 exists and whether or not it is busy, i.e. it will check MODEM1 and then MODEM2 and so on.

Though it is not recommended for normal usage, access to a specific modem can be "forced" using the /F switch, as illustrated in the following command line:

```
COMM/F MODEM USING SCRIPT 
```

As specified in the "modem access rules" above, the INI.TIW file is used by VersiCOMM to determine whether or not it is OK to take a terminal away from a job that is waiting for input. For example, it is common to leave a security program running on a job that is attached to a modem. This program handles the screening of incoming callers. Call the program LOGON. Most of the time LOGON just "sits there" waiting for a caller to input a password. In this situation it is acceptable to take the modem away from LOGON, use it and then give it back later. If, on the other hand, a remote caller was using the computer to place an order and the ORDER program was prompting the user to "Enter quantity:" you would definitely NOT want VersiCOMM to interfere. LOGON would be listed in the INI.TIW file, but ORDER would not be.

The INI.TIW file resides in DSK0:[7,0]. The VUE text processor is used to create the file. Program names are listed one name per line, with a maximum of six characters per name.

## VersiCOMM Initialization - The INI.TLK File

The INI.TLK file is a special kind of USING file that can be used to override the default values of the keyboard characters used to control VersiCOMM and to configure other operational parameters. It is created using the AlphaVUE text editor. Only VersiCOMM commands should appear in the INI.TLK file.

When VersiCOMM first executes it checks the user account to see if the file INI.TLK exists, if none is found it also checks DSK0:[7,0]. If found, the VersiCOMM commands in the file are processed. As is the convention in other types of USING files, all VersiCOMM commands must be prefixed with a "command character." The default command character is the dollar-sign, "\$", but this may be redefined as well by using the VersiCOMM command, COMMAND.

The following INI.TLK file would change the "change mode" character to a "\", the "direct save" character to a <Control-B>, the "break" character to a "#", and the "direct exit" character to an "&". It also enables input flow control using FLOW, turns JWAIT ON to conserve CPU time, and sets the IDLEOUT exit to 10 minutes.

```
;Example of an INI.TLK file
;
$CHGMODE 92
$DIRSAVE 2
$BREAKCHR 35
$DIREXIT 38
$IDLEOUT 600
$FLOW ON
$JWAIT ON
```

Most VersiCOMM commands that accept ASCII code numbers will also accept a literal character. The preceding example could have also been written:

```
;Example of an INI.TLK file
;
$CHGMODE \
$DIRSAVE ^B
$BREAKCHR #
$DIREXIT &
$IDLEOUT 600
$FLOW ON
$JWAIT ON
```

If RCLOCK is installed, it is recommended that the following commands be placed in the INI.TLK file:

```
$FLOW ON
$JWAIT ON
```

## VersiCOMM Initialization - The INIARG.TLK File

The INIARG.TLK file is similar in operation to the INI.TLK file, but it has a different purpose. The INI.TLK file is a static initialization file that remains (more or less) the same each time VersiCOMM is executed. The INIARG.TLK file is a dynamic initialization file



that is used to set the value of the VersiCOMM arguments ARG0-ARG15, and flags FLG0-FLG15, and to set or configure other VersiCOMM switches and options. The INIARG.TLK is session-oriented, i.e. it is intended for use during a particular session of VersiCOMM. In fact, VersiCOMM erases the INIARG.TLK file after it is processed.

Whenever VersiCOMM is executed it checks to see if there is an 'INIARG.TLK' file in the user account. If there is, VersiCOMM executes it, then erases it. Usually the INIARG.TLK file is created on the fly in another program which then chains to VersiCOMM.

Following is an example of an INIARG.TLK file that sets the value of several VersiCOMM arguments and flags.

```
;Example of an INIARG.TLK file
;
$SETARG 0 This is the first argument
$SETARG 1 This is the second argument
$SETFLG 0 127
$SETFLG 1 255
```

## Conversational Mode

When VersiCOMM is executed it will initially be in the Conversational mode. The following will be displayed:

```
VersiCOMM Version 3.x(xxx)
(C) 1984-1995 Valley Programming Service, Inc. - All Rights Reserved
Entering conversational mode...
```

Characters typed on the keyboard will be transmitted to the communications port, and data received by the communications port will be displayed on your terminal. If you are connected to a remote computer via a modem then to execute commands on the remote computer just type them in as if the computer were in the next room.

## VersiCOMM Status Line

If your terminal has a bottom status line and supports terminal function call 54, VersiCOMM will display a status line comprised of the following fields:

A	B	C	D	E	F	G
---	---	---	---	---	---	---

The status line display may be disabled by using the command \$STATUS OFF in the INI.TLK initialization file.

*Field A* is the VersiCOMM mode, it shows whether VersiCOMM is in COMMAND mode, CONVERSATIONAL mode, or is processing a SCRIPT.

*Field B* is the name of the script file, if any, that is currently being processed. This includes modem control files such as MODEM1.DL0, which are special cases of script files.

*Field C* is the baud rate of the modem port being used by VersiCOMM.

*Field D* displays call progress info, including:

ATTENTION OK	Modem was responsive.
NO RESPONSE	Modem was not responsive.
BUSY	Number was busy.
MODEM TIMEOUT	Modem timed out prior to achieving a CONNECT.
COMM TIMEOUT	WAIT for CONNECT message timed out.
COMM ABORT	Call aborted by a Control-C.
CONNECT speed	CONNECT achieved at speed indicated.

When a WAIT command is being processed Field E displays the word "WAIT" followed by the number of seconds remaining in the WAIT. Field E also displays the word "PAUSE" when a PAUSE is being processed.


*Field F* shows whether or not error correction has been requested when a number is dialed.

*Field G* shows the current state of the capture to disk switch, SAVE ON or SAVE OFF.

A snapshot of a typical status line while dialing might look like:


SCRIPT	MODEM1.DLL	9600	WAIT CONNECT	WAIT 004	REQ EC	SAVE OFF
--------	------------	------	--------------	----------	--------	----------

## Saving Displayed Information

To save the information that will be displayed on the terminal, key a . Note that you can toggle the SAVE function directly while in Conversational mode or you may use the "SAVE" command in Command mode. In Command mode you may optionally specify the file in which to save the incoming text.

Once the SAVE feature is enabled (by either method) it may be turned off to skip unwanted text and then turned back on to add additional text to the same output file. When SAVE is ON, captured text will be displayed in reduced intensity, UNLESS the remote system sends a terminal commands that turns reduced intensity off. The VersiCOMM status line and command mode status display also indicate the current state of the SAVE switch.

Multiple output files may be specified by CLOSEing the current output file and then using SAVE to specify a new output filespec.

The key that toggles SAVE ON and OFF, , may be redefined using the DIRSAVE command in VersiCOMM command mode, an INI.TLK initialization file, or a USING file.

## Recording Displayed Information

The RECORD command is used in combination with the SAVE command, when an emulation is in effect, to produce recordings of output from remote systems that can be played back in simulated real-time using a player written for the emulation.

This feature can be used to record demonstrations that can be played back on other systems.



For example, you could RECORD an Alpha Micro session using the GENeric emulation and then play it back using the player SHWGEN.

RECORD is only effective when SAVE is ON, and an emulation is enabled.

## Generating a 'BREAK'

A 'BREAK' is not a ASCII character. It is a CONDITION where the interface port transmits nothing but the equivalent of binary level 0 for an extended number of character times. No start bits, no stop bits, just what is called "spacing."

Although many communications and information services appear to be avoiding the use of a 'BREAK' for control purposes, it can still be a useful feature from time to time. For example, on EasyLink a message can be aborted using a 'BREAK', and on the Knowledge Index a search that is in progress can be interrupted.

Keying a @ while in Conversational Mode will generate a 'BREAK' condition on the port of a supported interface. The "Break" character, @, may be redefined using the BREAKCHR command in VersiCOMM command mode, an INI.TLK initialization file, or a USING file.

To use this feature your modem must be capable of detecting and transmitting a 'BREAK' condition and the interface must support generating a 'BREAK.'

## Changing to VersiCOMM Command Mode

Keying a <Control-^>, generated by the <HOME> key on many terminals, will switch the user from Conversational mode to Command mode. Keying it again will switch the user back. The "Change Mode" character <Control-^>, may be redefined using the CHGMODE command in VersiCOMM command mode, an INI.TLK initialization file, or a USING file.

## Direct Exit from Conversational Mode


Sometimes it is desirable to exit VersiCOMM directly from Conversational mode, rather than having to switch to Command mode and type QUIT. To do this a <~>, ASCII 126, is keyed. The "Direct Exit" character, <~>, may be redefined using the DIREXIT command in VersiCOMM command mode, an INI.TLK initialization file, or a USING file.

## Preserving the Conversational Mode Screen

If your terminal supports the Save Screen Area function, such as the AM65A, you can now use the PRESERVE command to set VersiCOMM so that when you switch from conversational mode to command mode and back again, the information displayed in conversational mode is preserved.

## Auto-Timeout From VersiCOMM

Using the IDLEOUT command, VersiCOMM can be set to automatically exit when there is no activity in Conversational or Command mode for the specified period of time. The format of the command is

```
>IDLEOUT time 
```

where 'time' is the number of seconds of inactivity that will trigger an automatic exit.

## Command Mode

### The Command Mode Status Display

When in command mode the following is displayed at the top of the screen:

VersiCOMM 2.1(385)									
On For		Time	Save	Echo	operational switches				
0 min		10:58 AM	OFF	OFF	AutoLF	TLocal	TRemote	Strip	JWait
					OFF	OFF	OFF	ON	ON
port selected									
Terminal	Driver	Interface	Port	Baud Rate	Parity	Size	Stop	DTR	
TAM1	PSEUDO	AM1000	0	19200	NONE	8	1	ON	
port configuration									
SERIO parameters									
Linefeed	Handshake	EOLChr	EOLTime	Emulate	BreakChr	ChgMode	DirSave	DirExit	
ON	OFF	LF	0.00	TTY	0			A	
control keys									
Current Login									
DSK1:[77.12]									
>									
COMMAND     19200         SAVE OF									

The values displayed are those at the time you switched to command mode. If a command is used to change the value of any parameter the status display will be immediately updated.

*With one exception (the BAUD command) the command that is used to change a particular parameter is spelled exactly as it is listed in the status display, i.e. to change the value of the "change mode character" the CHGMODE command is used.*

The parameters and switches of the status display are grouped into several categories.

#### Operational Switches

These are switches that directly affect the general operation of VersiCOMM.

##### Save

If this switch is ON information received by the terminal, and displayed on the screen, is also being saved in a disk file. The SAVE command is used to control this switch.

##### Echo

If ON, VersiCOMM will display (echo) characters as they are typed on the keyboard. Normally the remote computer (host) will echo the characters you type, so the default setting of this switch is OFF. This switch is changed by using the ECHO command.

##### AutoLF

If this switch is ON linefeeds are automatically appended to any carriage return typed via the keyboard or transmitted via an Answerback request. The AUTOLF command can be used to change this switch.

##### TLocal

This is the current state of the switch that enables translation and filtering of local characters, i.e. characters that you type on your keyboard, or send to the host in USING files

or conventional files. The TLOCAL command turns this switch on and off. The LOCAL command is used to set up the translation or filtering of a particular character.

### **TRemote**

This is the current state of the switch that enables translation and filtering of remote characters, i.e. characters that are received from the remote computer. The TREMOTE command turns this switch on and off. The REMOTE command is used to set up the translation or filtering of a particular character.

### **Strip**

This switch determines if characters should be stripped to 7 bits upon receipt from a remote source, before being used for any other purpose by VersiCOMM, including character translation, filtering, being displayed, or if the SAVE switch or RECEIVE is active, saved in a disk file. The STRIP command is used to control this switch. The default STRIP state is OFF.

### **JWait**

If JWait is ON VersiCOMM Conversational mode consumes no CPU time unless a character is typed at the keyboard or received at the communications port. The default JWAIT state is OFF.

### **Emulate**

Shows the terminal emulation currently in effect. Unless otherwise specified, VersiCOMM does not modify the control sequences of the terminal from which it is running or the control sequences of the remote host. In effect, VersiCOMM "disappears" and it is as if the terminal running VersiCOMM is connected directly to the host. When in this standard mode, VersiCOMM displays "TTY" under "Emulate" in the Command mode status display. "VT1" indicates that the VT100 terminal emulation is enabled, "GEN" that the Generic terminal emulation is enabled.

### **Port Selected**

These are parameters associated with the terminal/port COMM selected for communications. The port selected could vary on systems that have multiple modems.

### **Terminal**

This is just the name of the terminal that is being used for communications.

### **Driver**

This is the terminal driver that is currently defined for the terminal being used for communications. When exiting from VersiCOMM the driver that was in use prior to the execution of VersiCOMM is restored.

### **Interface**

This the interface driver that handles the hardware level I/O of data.

**Port**

A numerical reference to specify which port on an interface board is being used.

**Port Configuration**

These values define the current configuration of the port that is in use.

**Baud Rate**

The current baud rate of the communications port. Note, this is NOT the baud rate of the terminal from which you are running VersiCOMM. The baud rate of your terminal does not normally need to be changed. If your modem does not support speed conversion, the communications port and the remote computer port must be at the same baud rate. The BAUD command is used to change the communication port's baud rate.

**Parity**

Displays the current parity setting for the terminal port being used for communications. Will display "-na-" if an extended function interface driver is not being used. The PARITY command is used to change it's value.

**Size**

Displays the current number of data bits defined for the terminal port being used for communications. Will display "-na-" if an extended function interface driver is not being used. The SIZE command is used to specify a new value.

**Stop**

Displays the current number of stop bits defined for the terminal port being used for communications. Will display "-na-" if an extended function interface driver is not being used. The STOP command is used to specify a new value

**DTR**

Displays the current value of the signal which, by convention, is used to produce a 'DTR' signal to the modem. Will display "-na-" if an extended function interface driver is not being used. The DTR command can be used to change this value.

**SEND Parameters**

These parameters affect the operation of the SEND, SENDEPX and SENDL commands.

**Linefeed**

If this switch is ON linefeeds in a sequential file are transmitted to the remote computer when using the SEND or SENDEPX command, otherwise linefeeds are filtered. The LINEFEED command can be used to change this switch.

**Handshake**

If this switch is ON, VersiCOMM will react to X/ON and X/OFF characters received from the remote computer while SENDING a file. Using the translation facilities of VersiCOMM other characters could also be used for flow control. The HANDSHAKE command controls this switch.

### **EOLChr**

This parameter shows the character that is used by SEND and SENDL to "detect" the end of a line in a text file. The EOLCHR command changes its value.

### **EOLTime**

This is the time SEND will pause after transmitting a line of text. The EOLTIME command changes its value.

### **Control Keys**

These are keys that are used to control VersiCOMM. The number of operations controlled in this manner has been intentionally limited for the same reason that control of VersiCOMM via function keys has not been implemented -- VersiCOMM needs to be as "transparent" as it can reasonably be so that remote applications can be run with out VersiCOMM control sequences "getting in the way."

### **BreakChr**

This is the character that is used to generate a 'BREAK' from the keyboard. A 'BREAK' can only be generated on ports when an extended function interface driver is being used. This character can be redefined using the BREAKCHR command.

### **ChgMode**

This is the character that is used to switch back and forth between Command and Conversational mode. When switching from Command mode to Conversational mode this need not be the first character typed in the command line. The CHGMODE command can be used to specify a character other than the default.


### **DirSave**

This is the character that is used to turn the SAVE switch on and off directly from the Conversational mode of VersiCOMM. This character can be redefined using the DIRSAVE command.


### **DirExit**

This is the character that is used to exit from VersiCOMM directly from Conversational mode. It can be redefined using the DIREXIT command.









## **Entering VersiCOMM Commands**



VersiCOMM Commands are typed at the ">" while in Command mode. Entered commands are executed when a  is keyed. If a command is not spelled correctly, or not defined, the error message "?What" will be displayed on the line below the command prompt.


Commands may be abbreviated, as desired, as long as the abbreviation would not be the same for another command. For example, the command "BAUD 19200" may be entered as "B 19200" or "BA 19200" or "BAU 19200".

The last three commands entered can be recalled by keying a .

The following keys can be used for editing commands that have been typed at the VersiCOMM Command mode prompt:

<DEL> or <RUB>	Delete the preceding character.
	Delete the current character.
	Erase from current character to the end of the line.
	Move to the beginning of the line.
	Move to the end of the line.
	Insert space at current cursor position.
	Move cursor left one character.
	Move cursor right one character.
	Recall previous command.

To obtain a list of VersiCOMM commands type "HELP" followed by a  at the command prompt. To obtain information regarding a specific command type "HELP" and the name of the command, followed by a .

To exit from VersiCOMM to the AMOS operating system or other controlling program, type "QUIT" or "Q" at the VersiCOMM command prompt followed by a .

A detailed description of all commands can be found in the COMMANDS chapter of this manual.


## VersiCOMMander File Manager

The VersiCOMMander File Manager provides a "point and select" method of working with files in VersiCOMM command mode.

To start VersiCOMMander enter the SELECT command at the VersiCOMM Command mode prompt:

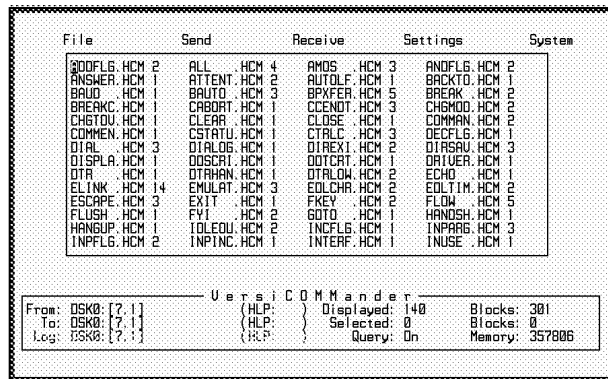
```
>SELECT 
```

If desired a wildcard file specification may be included in the command. For example:

```
>SELECT *.BAS 
```

The VersiCOMMander screen will appear similar to the following:





## The Main Window

The VersiCOMManDer Main Window is a scrollable list of files. The following key commands are defined when working in the Main Window:

	Move cursor, scrolling window as required.
<Page Down>	Scroll files down a page.
<Page Up>	Scroll files up a page.
	Exit from VersiCOMManDer.
<SPACEBAR>	Select or deselect file at cursor.
<ESC>	Display or hide current submenu.
	Display the corresponding submenu.
	Specify file set.
	View the file at the current cursor position.
	Hash selected files.
	Apply a LIT command to selected files.

## Selecting Files

Items may be selected by positioning the cursor on the item and pressing the spacebar. If already selected the item will be de-selected. If no items are selected the item on which the cursor is currently positioned is implicitly selected.

Selected items are displayed in reduced intensity. A count of selected items selected is maintained in the VersiCOMManDer Status Window. A count of the number of AMOS blocks contained in the selected files is also displayed.

All the items currently displayed may be selected using the Set All function in the File submenu, or de-selected using the Clear All function.

Once a set of items has been selected, a command may be applied to the entire set. For example, if a set of AlphaBASIC program files had been selected they could be transferred, hashed, etc.

It is not necessary to select all the displayed items. You may:

- Select the items one at a time by positioning the cursor on the desired item and pressing the spacebar, or
- Select all the items by using Set All, and then clear one or more items by positioning the cursor on the item and pressing the spacebar.

Any selected item can be cleared by positioning the cursor on the item and pressing the spacebar.

## The Status Window

The Status Window displays the following information:

"From:" is the currently defined Source account. If an ersatz name is defined for this account it is displayed in parentheses following the account spec.

"To:" is the currently defined Destination account. If an ersatz name is defined for this account it is displayed in parentheses following the account specification.

"Log:" is account you are currently logged into. If an ersatz name is defined for this account it is displayed in parentheses following the account spec.

"Displayed:" is the total number of items (i.e. devices, accounts, files or blocks) available for display in the Versi *COMMan*der Main Window. If Versi *COMMan*der is at the File level "Blocks:" will list the total number of blocks for all the displayed files.

"Selected:" is the total number of items, of those displayed, that have been selected for possible additional processing via hot keys or sub-menus. "Blocks:" lists the total number of blocks for all the selected files.

"Query:" is the current state of the Query switch. If Query is On the user is prompted to confirm most Versi *COMMan*der operations on an item by item basis.

"Memory:" is the total memory currently available to Versi *COMMan*der. This will vary depending on the number of items displayed in the Main Window and several other factors.

## File Submenu


### Specify


Displays the SPECIFY FILES dialog box. Enter the wildcard specification of the set of files you want to display.

Hot key: . Query: No.

### View


If no files are selected, views the file at the cursor. If one or more files are selected views each selected file.

When invoked by the  hot key, views only the file at the cursor, even if other files are selected.

Hot key: . Query: Yes.

### Hash

If no files are selected, calculates the hash total of the file at the cursor. If one or more files are selected calculates the hash total for each selected file.

Hot key: . Query: No.

### Copy

If no files are selected, copies the file at the cursor to the current destination. If one or more files are selected copies the selected files to the current destination.

If the destination is the same as the source, Versi *COMManDer* will prompt for a new file specification. Note that the destination is set from the Settings submenu.

Hot key: None. Query: Yes.

### Rename

In the RENAME dialog box enter the wildcard output file specification.

If no files are selected, renames the file at the cursor. If one or more files are selected renames each selected file.

Hot key: None. Query: Yes.

### Erase

Erases each selected file. A file must be selected to be erased and you must answer "Y" to the following prompt:

```
Do you really want to ERASE ALL selected files? Y/N:
```

Hot key: None. Query: Yes.

### Set All

Selects all files in the current file set.

Hot key: None. Query: No

### Clear All


Deselects all files in the current file set.

Hot key: None. Query: No

### Enter Command

Pops up the ENTER LIT COMMAND dialog box. Enter the name of the LIT file to apply to the selected files.

If no files are selected, applies the command to the file at the cursor. If one or more files are selected applies the command to each selected file.

Hot key: . Query: Yes


**Exit**

Exit Versi*COMM*ander.

**Send Submenu****Enter Command**

Pops up the ENTER LIT COMMAND dialog box. Enter the name of the LIT file to apply to the selected files.

If no files are selected, applies the command to the file at the cursor. If one or more files are selected applies the command to each selected file.

Hot key: . Query: Yes

**FasTrans**

Executes the TRANS file transfer program.

Hot key: None. Query: Yes

**Zmodem**

Executes the SENDZ file transfer program.

Hot key: None. Query: Yes

**Ymodem**

Executes the SENDZ file transfer program with the /Y switch applied.

Hot key: None. Query: Yes

**Xmodem**

Executes the SENDX file transfer program.

Hot key: None. Query: Yes

**Xmodem 1K**

Executes the SENDX file transfer program with the /Y switch applied.

Hot key: None. Query: Yes

**B-Plus**


Executes the BPXFER file transfer program. Used primarily with Compuserve.

Hot key: None. Query: Yes

**Receive Submenu****Enter Command**

Pops up the ENTER LIT COMMAND dialog box. Enter the name of the LIT file to apply to the selected files.

If no files are selected, applies the command to the file at the cursor. If one or more files are selected applies the command to each selected file.

Hot key: . Query: Yes

### **FasTrans**

Executes the RETRV file transfer program.

Hot key: None. Query: Yes

### **Zmodem**

Executes the RECZ file transfer program.

Hot key: None. Query: Yes

### **Ymodem**

Executes the RECZ file transfer program with the /Y switch applied.

Hot key: None. Query: Yes

### **Xmodem**

Executes the RECX file transfer program.

Hot key: None. Query: Yes

### **Xmodem 1K**

Executes the RECX file transfer program.

Hot key: None. Query: Yes

### **B-Plus**

Executes the BPXFER file transfer program. Used primarily with Compuserve.

Hot key: None. Query: Yes

## **Settings Submenu**

Switches are toggled when selected. If ON, it is turned OFF. If OFF, it is turned ON.

### **Query    ON**

When ON, will query file operations such as Copy, Rename, Erase and file transfers. Default is ON.

### **Delete   OFF**

When ON, applies the /DELETE switch to file transfers, if the switch is defined for the protocol that has been selected. Default is OFF.

### **IND File   OFF**

When ON, file transfer protocols that accept wildcards will use the "indirect" file feature of the AMOS wildcard system. If FasTrans is selected from the Send Submenu, file list processing is used if this switch is ON. Default is OFF.

**Erase Ctrl ON**

When ON, asks the user to confirm the use of the ERASE command. Default is ON.

**Destination**

Enables the user to select the destination for a COPY command.

**System Submenu****System Status**

Executes a STAT command.

**Network Status**

Executes an NSTAT command.

**System Info**

Executes a SYSTEM command.

**Terminal Defs**

Executes a TRMDEF command.

**Ersatz Names**

Executes an ERSATZ command.

**Translating & Filtering Characters**

As described in the overview, VersiCOMM has the capability to translate local characters you type on a keyboard or send in a text file, before sending them to a remote computer, and it also can translate characters received from a remote computer before using them locally. An entry in the local or remote translation tables is made using the LOCAL or REMOTE commands. Translation and filtering can be enabled or disabled without destroying the existing tables using the TLOCAL and TREMOTE commands. FLUSH can be used to clear one, or both, translation tables.

Any 8-bit code can be translated into any other (non-null) 8-bit code, OR any 8-bit code can be filtered, i.e. ignored. The allowable decimal values for these codes are 0 through 255. For instance, the command "LOCAL 127,8" will convert a del (rubout) to a backspace before it is sent to the remote computer. The command "LOCAL 127,0" would discard any del rather than sending it to the remote computer. Specifying one argument instead of two, as in "LOCAL 127", will result in the system telling you what action, if any, is being taken with local character 127. For example,

"Local character 127 is not converted!"

"Local character 127 is converted to 8 before sending to host"

"Local character 127 is discarded before sending to host"

An entire character set could be translated by setting up the LOCAL and REMOTE translation tables, by executing appropriate LOCAL and REMOTE commands. This

commands need not be re-entered one by one each time you call the computer requiring translation, but may be specified as part of the initialization process in a unique USING file.

A detailed description of the commands used in translation and filtering will be found in the COMMANDS chapter of this manual.

## Filtering Nulls

The filtering of nulls received from the remote computer is a special case. *If remote translation is enabled and as least one non-null remote character is being filtered or translated, nulls will be automatically filtered. More specifically, if the remote translation table has been allocated, and TREMOTE is ON, then nulls will be filtered. At least one filter or translation must be specified for the table to be allocated.*

If you wish to filter ONLY nulls, a filter or translation must be specified and then removed by a sequence such as the following:

```
$TREMOTE ON      ;Enable remote translation/filtering
$REMOTE 1,0      ;Set arbitrary remote filter
$REMOTE 1,1      ;Restore to original character
```

This would leave the translation table allocated, and only nulls would be filtered.



### APPLICATION NOTE TRANSLATING & FILTERING HIGH CODES

Some services set the high (unused in ASCII) bit of a character to 1, or they use the bit for parity, which results in the high bit being set on some characters but not on others. If STRIP is set ON, you need not worry about adding 128 (the value of binary 10000000) to the ASCII value of the character in order for it to be filtered. STRIP masks out this bit BEFORE checking to see if the character should be translated or filtered.

## The AMOS Command

The AMOS command (see COMMAND Chapter) enables VersiCOMM to execute AMOS commands without leaving VersiCOMM. This means that you can do a DIR while still in VersiCOMM, or TYPE a file, or do a SYSTAT, etc. Also local computer commands may be used in the USING file in addition to remote computer and VersiCOMM commands! This means that BASIC or other programs can be executed under the control of a USING file. Certain restrictions apply to the use of this command. See the COMMANDS chapter for additional information.

Common uses of AMOS include using the "XY" command in USING files and checking the time using TIME when online.


## Command Files, Task Manager & Shell

VersiCOMM may be run as part of a sequence of programs in a command file. VersiCOMM can also be run as a scheduled task under the control of the Task Manager.


VersiCOMM has been used extensively in Alpha Micro's Shell Menus. A sample Shell Menu, the COMMCENTRE, is included with AMOS/L VersiCOMM. This Menu illustrates how VersiCOMM can be used to perform a number of different communications services from a Menu.

## The COMMCENTRE

The COMMCENTRE is an AlphaMENU based user interface that acts a "front end" to VersiCOMM. The intent being to take a particular communications task, such as logging on to Compuserve, and automate it as fully as possible.

To run the COMMCENTRE, at the AMOS dot prompt type "CENTRE" (or "CENTER") followed by a . A menu similar to the following will be displayed:

*** UPS COMMCENTRE ***		
Monday, May 1, 1995 07:50:58 PM		
Mail/Message Services	Information Services	Information Services
11 MCI Mail	21 Knowledge Index	31 Lexis/Nexis
12 AT&T EasyLink	22 Dow Jones News/Ret.	32 WestLaw
13 Graphnet	23 AT&T FV1	33 DIALOG
14 AT&T Mail	24 Compuserve	34 Genie
	25 NewsNet	35 Delphi
Internet Access	Link to Remote Computer	Other Services
41 Netcom	51 Access Modem	61 Calendar 1900-2100
	52 DIALER	62 VUE a File
	53 TYMNET Access Info	63 PRINT a File
	54 SprintNet Access Info	64 AMOS Shell Menu
		65 SETUP Service Access
		66 EXIT to AMOS
Enter menu selection (type "?" for help): 100		

Selections from the menu are made by using the cursor keys to position the "window" over the desired function, or by entering the functions selection number, and then keying a .

*Most COMMCENTRE functions require that you have previously "setup" service access by selecting function 65 of the COMMCENTRE Main Menu and using the appropriate Access Parameter Definition program to enter the access numbers and logon data for the services you wish to access. See the VersiCOMM I INSTALLATION MANUAL for more information on COMMCENTRE Access Parameter Definition.*



It is not necessary to use the AlphaMenu system to initiate COMMCENTRE functions. Nearly all COMMCENTRE functions may be initiated via a command file. As a result, a function such as "Access MCI Mail" can be performed from the COMMCENTRE menu, from AMOS by entering the command "MCI", or from other software that chains to the command "MCI". This provides a great deal of flexibility in the way VersiCOMM can be used to communicate. For example, users have written ESP GLUE menus that replace the AlphaMenu system.

## Dialing Methods

In addition to the COMMCENTRE, there are several ways a remote system can be dialed using VersiCOMM. In addition to direct dialing, the modem can be dialed using the VersiCOMM DIAL command, a series of VersiCOMM commands in a TLK file, or using the screen DIALER.

### Dialing in Conversational Mode

If you prefer, you can bypass all of VersiCOMM's dialing systems and talk directly to the modem in conversational mode. The modem can then be dialed by entering a dial command string in the modem's command language. For example, entering "ATDT555-5555."



## VersiCOMM DIAL Command

The DIAL command can be used in VersiCOMM command mode to dial a number specified in the command line, or a number that is contained in a file specified in the command line. For example:

```
>DIAL 555-1212      OR      >DIAL SITE1.PHN
```

## VersiCOMM Script File

If a number is dialed frequently a script file can be written to access and log on to the remote system. The script file can then be invoked via an AMOS command or menu system. See the chapter on USING files, i.e. script files, for more information.

## The DIALER

The DIALER is a fully integrated system that enables the user to dial, connect, log onto, and off of most computers and services without having to write script files. It includes an easy to use fill-in-the-blank database, and a special modem control file generator. Please refer to the DIALER chapter for more information.

## MLON and MLOFF

MLON and MLOFF are AMOS level commands that are intended to be used by the remote user of an Alpha Micro to prevent access to a modem while the remote user is on the system. They may be incorporated into a remote user logon command sequence for convenience.

When executed, MLON marks the terminal it is being run on as being "in use," and outputs the message:

```
%Access lock is now ON
```

MLOFF frees the terminal for communications and outputs the message:

```
%Access lock is no OFF
```


Note, MLON and MLOFF are NOT security programs, they just prevent a local user from "stepping on the toes" of a remote user while the remote user is busy on the system.

## MATCH

MATCH is an AMOS level command used on a modem (or other) job for automatic matching to a connected terminal's baud rate. Baud rates supported include 300, 1200, 2400, 9600, 19200 and 38,400 bps. An optional "system access" password can be defined, and a command can be specified in the MATCH command line that will be executed after the baud rate has been established and system access granted.

## Operation

The format of the MATCH command is:

```
.MATCH {/L}{/X}{Commandspec} 
```

where Commandspec is a command to be executed when MATCH has matched the baud rate, and the user has successfully entered the system password, if defined.

MATCH must be run on the job that is attached to the terminal to be matched. If, for example, you have a 2400 bps modem connected to the port defined as MODEM1 and JOB6 is attached to MODEM1, then JOB6 must be running MATCH if it is to match baud rates with the modem. This can be done in several ways. One way is to FORCE the job to run MATCH from another job. For example,

```
.FORCE JOB6 MATCH LOGON
```



Another is to incorporate the MATCH command into a LOGOFF sequence that users initiate when they logoff your system.

MATCH can also be incorporated into the system initialization file for the job on which it is supposed to run. For example:

```
ATTACH MODEM1,JOB2      ;Set up and enable JOB2 with
KILL JOB2                ;32K of memory.
FORCE JOB2
MEMORY 32K
VER
LOG SYSTEM SERVICE
MATCH RUN SECURE
```

An optional "system access" password may be specified by using VUE to create the file DSK0:MATCH.DAT[7,0]. The first line of this file is used as the password.

Note that the "system access" password is not intended to be a logon password for a specific user, but rather a first line of defense to unauthorized access. MATCH could easily chain to a user specific logon program once general access had been granted. The password file need not be encrypted since anyone that can VUE the file already has access to the system.

When trying to match baud rates users should press the  key to allow rate determination. MATCH will typically match rates in the following number of RETURNS. The number required will be higher if the  key is pressed too quickly, or there is noise on the line.

BAUD RATE	RETURN's	with /L	with /X	with /L/X
38400	3	NA	3	NA
19200 bps	2	NA	2	NA
9600	1	NA	1	NA
4800	NA	NA	2	2
2400	2	1	3	1
1200	3	2	4	4
300	4	3	5	5

Fewer RETURNS are required at low-speed baud rates when the /L switch is specified.

MATCH is programmed so that no CPU time is consumed when it is idle. MATCH will revert to the idle state if the port becomes inactive during the matching process.

## Switches

The following switch may be specified:

- /L Specifies that only low-speed baud rates will be matched. This includes 300, 1200, and 2400 bps. This reduces the number of RETURNS that must be entered to match rates. If /X is specified, 4800 bps will also be matched, but the number of returns required increases.
- /X Specifies that 4800 bps should also be matched.

## Errors

### ?PIC file read error

The program has not been correctly keyed to the Software Security Device.

### ?Bad password

The password entered does not match that specified in DSK0:MATCH.DAT[7,0].


### ?Input timed out

The time allowed for entry of the password expired.

## TRMSTS

TRMSTS is an AMOS level command that can be used to check the state of the terminal status bits. In an emergency you can also reset various status bits. Note that in normal operation VersiCOMM should not exit and leave, for example, the "terminal assigned" bit on. This can only occur if there is an abnormal exit. Also note that VersiCOMM can give you a "terminal in use" message for several reasons, not just if the terminal assigned bit is set.

The format of the TRMSTS commands is:

```
TRMSTS {terminal-name} 
```

If terminal-name is omitted, MODEM1 is assumed. You must be logged into OPR: to run TRMSTS.

When run, TRMSTS generates a table similar to the following:

```

Terminal Status -- MODEM1
-----
T$IMI - image mode - not set
T$ECS - echo suppress - * set *
T$LCL - local mode - not set
T$DAT - data mode - * set *
T$ILC - input lower case - not set
T$XLT - translate fkey - not set
T$NFK - suppress fkey - not set
T$OIP - output in progress - not set
T$LED - line editor - * set *
T$ASN - terminal is in use - not set
T$DIS - terminal disabled - not set
T$VLD - position valid - * set *
T$LDT - special line edit - not set

T.JLK - terminal to job link - JOB2
JOBTRM - job to terminal link - MODEM1
JOBPRG - program on trm's job - MATCH (running)

Clear T$ECS status bit?

```

It then asks if selected bits should be cleared. If you want to clear a bit you must respond with a "Y".


If the terminal is linked to a job but not attached to the job, it will also ask you if you want to clear the terminal to job link.

## CHGTDV

On AMOS 2.x systems you may use the AMOS command SET TERMINAL DRIVER to change the terminal driver "on the fly." CHGTDV can be used on systems, such as AMOS/L 1.3D and 1.4, that do not support this capability.

### Operation

The format of the CHGTDV command is:

```
CHGTDV {terminal-driver} 
```

If terminal-driver is omitted CHGTDV will display the drivers that are available in the Terminal Definition Chain or in System Memory. For example:

```

USAGE: CHGTDV terminal-driver

Current terminal driver is GEN

Terminal drivers available in terminal definition chain:

TVI      AM65      TELTYP  GEN      WINDOW  TVI925


Terminal drivers available in system memory:

AM70      AM75

```

Otherwise, CHGTDV checks that the available impure area, if defined, will be adequate for the new driver. If it is, the driver is changed and the following message is output:

```

.CHGTDV GEN 
Terminal driver changed to GEN.TDV
Call driver initialization routine?:

```

If the new driver has an initialization routine you will be asked if it should

be run.

## Errors

### ?Can't change terminal driver -- job is a slave task

CHGTDV won't change the terminal driver of a slave task, such as the WINDOW driver of a MULTI job.

### ?Can't find 'terminal-driver'

The terminal driver specified cannot be located in the terminal definition list, or in system memory.

### ?Cannot use this driver -- impure area overflow

The impure area that is available is not adequate for the needs of the requested driver.

### ?Cannot use this driver -- not a 'real' driver


The driver requested has the TD\$NTD bit set.

## Filter

FILTER is included in the VersiCOMM package to provide additional flexibility in the filtering and translation of files received from remote sources such as EasyLink or Lexis.

In its standard format it filters all "unprintable" characters from a file. In addition it prepends a carriage return to any isolated linefeed and appends a linefeed to any isolated carriage return. It saves a .BAK copy of the original file.

The format of the FILTER command is:

```
FILTER filespec 
```

The /W:n switch is used to force a line wrap after n characters. For example /W:80 would wrap lines longer than 80 characters.

The M68 source for FILTER will be found in release account [7,121].



## The DIALER

---

### Overview

VersiCOMM's DIALER system is comprised of three major components:

1. The primary DIALER screen, used to select and dial DIALER database entries and to access other DIALER system screens.
2. The DIALER database Entry Screen, used to add, delete, and modify entries.
3. The DIALER Modem Setup Screen, used to specify the various parameters associated with a particular modem, to initialize a modem using a specified string or script file, and to build a modem's attention, dial, and hang-up control files.

To start the DIALER enter the command "DIAL" and a  from the AMOS command level, or setup your menu system to include the DIAL command.

### DIALER Data Files

DIALER uses two data files. DSK0:VPSMDM.DAT[7,121] is used to store modem data for each modem on the system, e.g. MODEM1, MODEM2, MODEM3, etc. Its default size is 15 records. DSK0:VPSDLR.DAT[7,121] is used to store DIALER entry data. Its default size is 50 entries. If not appropriate for your system the defaults may be changed by modifying the value for SETUP'SIZE and DLR'SIZE in DIALR5.BAS and recompiling the program.

### DIALR5.BAS Source File

The AlphaBASIC source file for the DIALER, DIALR5.BAS, is Z/Archived in the self-extracting release file SRCBAS.LIT[7,121]. This file is provided solely for the convenience of licensed users of VersiCOMM-Plus, so those qualified may modify or correct the operation of the DIALER according to their own needs.

### Function Key Operation

DIALER screens expect function keys to generate AM62A compatible keycodes. If your terminal is not compatible, a keyboard translation file, trmdef.VPX, must be created that translates function keys to produce AM62A compatible keycodes. The DIALER requires the

trmdef.VPX file to be loaded in system or user memory. Note that 'trmdef' refers to the terminal definition name of your terminal, e.g. AM65A, WYSE50, etc.

A list of function key definitions will be found in the file DIALR5.BAS. The program VPSTRN can be used to create the translation file. Using VPSTRN, the decimal value of a function key translation may be input directly. See the Terminal Emulation chapter of this manual for more information on the operation of VPSTRN.

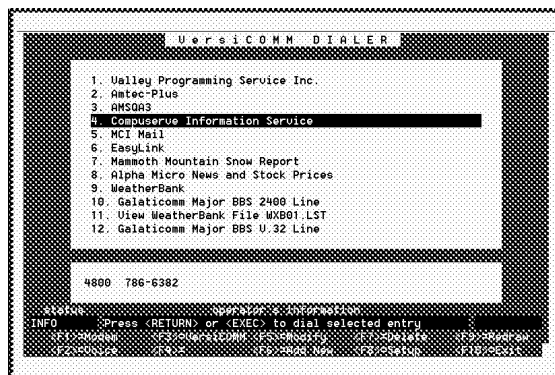
## DIALER Subroutines

The DIALER program uses several assembly language subroutines. For maximum efficiency two of these subroutines, VPS200.SBR and VPS300.SBR should be loaded in system or user memory.

## The Dialer Screen

The DIALER Screen lists 12 DIALER entries per page. The selection bar will be positioned over the first entry to start. The port baud rate for the current selection and its phone number (or file) will be displayed in the selection data window. Entries which have not yet been used will be listed as "available."

Operator's information is displayed in a special window near the bottom of the screen, above the function key definitions. An example of the DIALER Screen follows.











## Selection Controls



The following keys, or key combinations, are used to position the selection bar and to select a particular entry:














	Moves the selection bar up one entry.
	Moves the selection bar down one entry.
<page-up> or	Displays the previous page of entries.
<page-down> or	Displays the next page of entries.









<HOME>	Displays the first page of entries and positions the selection bar on the first entry.
 	Positions the selection on the last entry and centers the last entry in the window.
 	Centers the current selection in the window.
 or <EXEC>	Dials the modem number of the selected entry. If there is no modem number specified for the entry and there is a voice number the voice number is dialed instead. If there is no modem or voice number an error message is displayed.
key-word 	Positions the selection bar on the first entry that contains the keyword or string specified. Will scroll through all entries until the keyword or string is found. To abort the search enter a   or <CANCEL>.
<ESC>	Exits the DIALER.


## Function Key Actions

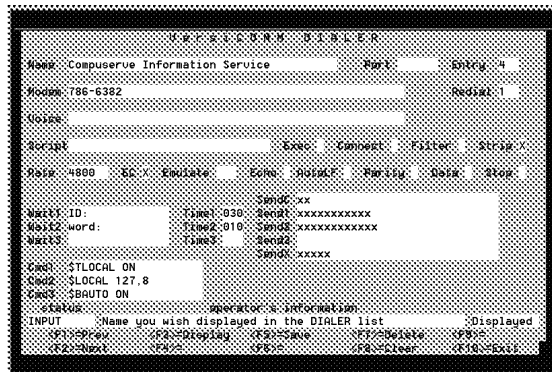
The following function keys can be used to initiate the actions indicated. Note that the numeric keys  to  serve as alternates to the function keys and can be used in their place.

 or  Dial Modem Number	Dials the modem number of the selected entry. If there is no modem number a message is displayed.
 or  Dial Voice Number	Dials the voice number of the selected entry.
 or  Execute VersiCOMM Directly	This selection executes VersiCOMM as if the command "COMM portname" had been entered. 'Portname' is the name of the modem specified in the 'Port' field of the Entry Screen. If 'Port' is not specified for the entry, then MODEM1 is used.
 Reserved	
 or  Modify Database Entry	This function switches to the Entry Screen and displays the data for the selected entry.
 or  Add New Database Entry	Switches to the Entry Screen and locates the next available record so the user may create a new entry.
 or  Delete Database Entry	Deletes the selected entry.

 or  Modem Setup Screen	Switches to the Modem Setup Screen.
 or  Redraw Screen	Refreshes the screen.
 or <F13> or  Exit	Exits the DIALER.

## Entry Screen

The Entry Screen is used to maintain dialing information for each DIALER entry. The user moves from field to field by pressing a  or other field terminating character. An example of the Entry Screen follows.



## Entry Screen Fields

A description of each of the fields on the Entry Screen follows. All fields but the Name field are optional.

**Name** - A description of the DIALER entry. May be up to 40 characters.

**Port** - The port (and modem) that should be used when this entry is selected. If the port is not specified, COMM will use the default port, MODEM1.

**Entry** - The entry number. This field is only specified by the user when a particular entry is to be displayed, or the existing values are to be copied to another entry.

**Modem** - The phone number of the modem for the DIALER entry. May be up to 50 characters.

**Redial** - The number of times to redial the modem number if a CONNECT is not achieved.

**Voice** - The phone number for voice communication for the DIALER entry. May also be up to 50 characters.

**Script** - The file specification for the VersiCOMM script file that should be executed according to the Exec and Connect fields. If neither the Exec field or the Connect field are marked, then the script file is ignored.

**Exec** - If marked with an "X" or other non-blank character indicates that the script file in the Script field should be executed instead of using the information in the DIALER entry record.

**Connect** - If marked with an "X" or other non-blank character indicates that the script file in the Script field should be executed when a CONNECT is achieved. Applicable information in the DIALER record is used to dial and configure the modem prior to achieving a CONNECT.

**Filter** - If marked with an "X" or other non-blank character indicates DIALER will execute the script DSK0:FILTER.TLK[7,121] to configure VersiCOMM to filter control characters. Generally, you should only invoke filtering when accessing a service as a "TTY" terminal, otherwise terminal specific control and escape sequences would be filtered out.


**Strip** - If marked with an "X" or other non-blank character indicates that VersiCOMM should "strip" the high (i.e. eighth) bit of each byte.

**Rate** - This is the rate the port will be set to when the modem is dialed.

**EC** - If marked with an "X" or other non-blank character indicates that the modem commands listed in the "with error control" column of the modem setup screen should be used in controlling the modem. *If your modem is set to use error control as a default it is not necessary to use this option, or this option can be used for other purposes.*

**Emulate** - Specifies the three character name of the emulation desired. For example, "VT1" for a VT100 emulation, or "GEN" for a VPS generic terminal emulation.

**Echo** - If marked with an "X" or other non-blank character indicates that the remote will not echo the characters you type at the keyboard, so VersiCOMM should echo the characters.

**AutoLF** - If marked with an "X" or other non-blank character indicates that VersiCOMM should append a line-feed to any  that you enter.

**Parity** - DIALER will configure the port for odd, even, or no parity, according to the specification of the letters O, E, or N.

**Data** - DIALER will configure the port for 7 or 8 data bits, according to the number specified in this field.

**Stop** - DIALER will configure the port for 1 or 2 stop bits, according to the number specified in this field.

**SendC** - A string, comprised of up to 30 characters, that will be transmitted to the remote when a CONNECT is achieved. If the string needs to be terminated with a carriage-return, a "\*M" must be included at the end of the string. The "\*" character is a prefix that indicates the next character is a control character. Following are some examples of prefixed characters:

String	ASCII Value	Character
*M	13	Carriage-return
*C	3	Control-C
*J	10	Linefeed
*j	106	Asterisk

After entry this field is displayed as lower cases x's.

**Send1, Wait1, Time1** - Send1 is string of up to 30 characters that is transmitted after the SendX string is sent. Wait1 is an optional field of up to 15 characters that is the string VersiCOMM should wait for after sending the Send1 string. Time1 is the time to wait for the Wait1 string. After entry the Send1 field is displayed as lower case x's.

**Send2, Wait2, Time2** - See the preceding description.

**Send3, Wait3, Time3** - See the preceding description.

**SendX** - A string, comprised of up to 30 characters, that will be transmitted to the remote when the user exits from VersiCOMM. If the string needs to be terminated with a carriage-return, a "\*M" must be included at the end of the string. The "\*" character is a prefix that indicates the next character is a control character. See the description of the SendC field for examples of prefixed characters. After entry this field is displayed as lower cases x's.












**Cmd1, Cmd2, Cmd3** - Are additional VersiCOMM script file commands that will be executed when VersiCOMM is initialized. These commands must be written exactly as they would appear in a script file. For example,

```
$TLOCAL ON
$LOCAL 127,8
```



If three commands is not enough then the \$SENDTLK command can be used here to execute a script file with as many lines as is needed.














## Field Editing

The following line editing functions are supported:

<RUB>	Delete previous character.
<CHAR INS> or 	Character insert.
<CHAR DEL> or 	Character delete.
	Move cursor to end of line.
	Move cursor to beginning of line.
	Erase to end of the line.
	Move right to next character.
	Move left to previous character.
 or  or 	Advance to next field.
	Move to previous field.

## Function Key Actions

The following function keys can be used to initiate the actions indicated. Note that by pressing the <ESC> key, followed by one of the numeric keys  to , you can also initiate the described action.

 or   Display Previous Entry	Displays the previous entry. If current entry is #1, then has no effect.
 or   Display Next Entry	Displays the next entry. If the current entry is the last entry in the file, then has no effect.
 or   Display Specified Entry	Displays the entry specified in the Entry field.
 Reserved	
 or   Save Displayed Entry	Saves the currently displayed entry. Note that the Entry field determines the record in which the data will be saved.



**Modem** - TRMDEF name of the modem port, usually MODEM1, MODEM2, etc.

**Attn Standard** - Command string that prepares the modem for use when error correction will not be used. At a minimum should turn on result codes and select verbose result codes. If modem is configured for error correction then you will also need to include the string that turns error correction off.

**Attn With Error Control** - Command string that prepares the modem for use when error correction will be used. At a minimum should turn on result codes and select verbose result codes. If modem is not configured for error correction then you will also need to include the string that turns error correction on.

**Attn Parameter** - Wait time for the modem to respond to the attention string. Would normally be about 3 seconds.

**Attn Response** - Modem's response to the attention string. Will usually be "OK".

**Dial Standard** - Command string to dial the modem without using error correction. It's a good idea to include the attention string as part of the dial sequence so that if the DIAL command is used without first using an ATTENTION command it will still work properly.

**Dial With Error Control** - Command string to dial the modem with error correction requested.

**Dial Modem Suffix** - String that terminates a dial command string. Will usually be a carriage-return, "\*M".

**Dial Voice Suffix** - String that returns the modem to command mode after a number is dialed. Will usually be ";\*M".

**Dial Parameter1** - So it can readily detect a BUSY, the DIAL command waits for a CONNECT for a few seconds, and then checks for a BUSY, then waits for a CONNECT for a few seconds, and so on. Dial Parameter1 is the amount of time it waits for a CONNECT each scan. Normally this value would be about 5 seconds.

**Dial Parameter2** - Is the number of times DIAL will wait for a CONNECT and then check for a BUSY response. A typical number of times is 8.

**Dial Connect Response** - The message the modem sends when a CONNECT is achieved. Will usually be "CONNECT".

**Dial Busy Response** - The message the modem sends when a busy signal is detected. Will usually be "BUSY".

**Escape Standard** - The sequence of characters that should be sent to the modem to cause it to "escape" to command mode, from the on line mode. Will usually be "+++".

**Escape With Error Control** - Provision is made for a different escape sequence when using error correction. Will usually be the same as the Standard escape sequence.

**Escape Parameter1** - The "guard" time that begins the modem escape sequence. Will usually be about 1.5 seconds.

**Escape Parameter2** - The "guard" time that ends the modem escape sequence. Also will normally be about 1.5 seconds.

**Escape Response** - The response of the modem when it has returned from online mode to command mode. Will usually be "OK".

**Hang-up Standard** - Command string that is used to hang-up the modem. Will normally also include a modem attention string to insure modem's response will be seen.

**Hang-up With Error Control** - Command string that is used to hang-up the modem when error correction has been requested.

**Hang-up Parameter1** - Standard wait time for modem to hang-up.

**Hang-up Parameter2** - Long wait time for modem to hang-up.

**Hang-up Response** - String that modem sends when it has successfully hungup.

**Reset Standard** - Command string that is used to reset the modem to its current set of default parameters.

**Reset With Error Control** - Command string that is used to reset the modem when error correction has been requested.

**Reset Parameter** - Wait time for the modem's response to being reset.

**Reset Response** - Modem's response to being reset. If modem is set (as a default) not to send result codes there will be no response.

**Init** - The command string, or alternatively, the file specification of the script file used to initialize the modem.




















## Field Editing

Field editing is the same as with the Entry Screen.

## Function Key Actions


The following function keys can be used to initiate the actions indicated. Note that by pressing the <ESC> key, followed by one of the numeric keys <1> to <9>, you can also initiate the described action.













 or  1 Initialize Modem	Initializes the modem specified in the Modem field using the command string or script file specified in the Init field.
 or  2 Build Modem Control Files	Builds the modem control files for the modem specified in the Modem field.
 or  3 Display Specified Entry	Displays the modem setup info for the modem specified in the Modem field.
 Reserved	
 or  5 Save Displayed Entry	Saves the modem setup info for the modem specified in the Modem field.
 Reserved	
 or  7 Delete Specified Entry	Deletes the modem setup info for the modem specified in the Modem field.
 or  8 Clear Screen Fields	Clears all screen fields.
 or  9 Fill Fields With Defaults	Fills screen fields with default values.
 or <F13> or   Exit	Exits and returns to the Dialer Screen.

## Dialer Example - Accessing CIS

The following example illustrates the use of the DIALER by setting up access to the Compuserve Information Service. To actually access the service you must have an account established with Compuserve, know your User ID, Password and local Compuserve telephone number. When you have this information available:

1. Access the DIALER using the DIAL command at the AMOS command level; or if the DIALER has been incorporated into your menu system, by making the appropriate menu selection.
2. From the DIALER Screen press the  function key to add a new entry.

3. In the name field type "Compuserve Information Service" and a .
4. If your modem's TRMDEF name is MODEM1, don't enter anything in the Port field, just enter a . If you are using a name other than MODEM1, enter the name in this field and press . Note that this is the name of the modem port TRMDEF, NOT the modem JOB.
5. At the Entry field just press  to leave it blank. This indicates a new entry.
6. Enter the local Compuserve phone number in the Modem field. For example, "5551212". In the Redial field enter the number of times you wish to redial if the number is busy or a connect is not achieved for some other reason.
7. Using the  or  keys move the cursor to the Strip field and type an "X".
8. Move to the Rate field and enter the port baud rate. If you are using a 2400 bps modem without compression the rate would normally be 2400, and with compression and speed conversion enabled would be 4800. If you are using a V.32 9600 bps modem, the rate would be 9600 or 19200.
9. The next field is the EC field. If you are using a modem capable of error correction and the modem is not preconfigured to use error correction mark this box with an "X".
10. Now move down to the SendC field. This string is transmitted when the modem has achieved a successful connection. In this field type "\*C" and then press . Note that DIALER converts this field to "xx" to mask its contents. The "\*C" has been stored.
11. In the Wait1 field enter "ID:", and in the Time1 field enter a "30". In the Send1 field enter your Compuserve ID, followed by a "\*M". For example, "55555,555\*M". This field is displayed as "xxxxxxxxxxxx".
12. In the Wait2 field enter "word:", in the Time2 field enter a "10", and in the Send2 field enter your password, followed by a "\*M". For example, "PASSWORD\*M". This field is also masked with x's.
13. Next skip down to the SendX field. This string is sent when you exit from VersiCOMM using "~" in Conversational mode or QUIT from Command mode. In the SendX field enter "BYE\*M".
14. Now move to the Cmd1 field and enter "\$TLOCAL ON", and in the Cmd2 field enter "\$LOCAL 127,8". Enter the command "\$BAUTO ON" in the Cmd3 field. Do NOT end these strings with a "\*M".
15. Press the  function key to save the entry and the  function key to return to the DIALER screen.

The Compuserve Information Service may now be accessed by positioning the selection bar over its entry and pressing .



If you have not generated your modem's control files using the DIALER's Modem Setup Screen, this will need to be done prior to using VersiCOMM or the DIALER. See the VersiCOMM INSTALLATION MANUAL for more information.

## Dialer Utilities

The self-extracting Z/Archive file SRCBAS.LIT contains the AlphaBASIC source code for the following DIALER utilities:

EXPDLR.BAS Can be used to expand or sort the DIALER data file VPSDLR.DAT.

LSTMDM.BAS Can be used to list the modems in the DIALER data file VPSMDM.DAT.




## 'USING' Files

---

### Introduction

As described in the OVERVIEW Chapter a "USING" file may be specified in order to initialize and or control the operation of COMM. A USING file is a text file that is created using VUE or other text editor. It may contain VersiCOMM commands or text that will be sent "on down the line" to your modem or remote computer. Since smart modems are normally designed to be operated from a terminal keyboard, they may be controlled by an appropriately written USING file. VersiCOMM can be executed under the control of a USING file by specifying a USING file specification in the COMM command line:

```
.COMM USING VPS 
```

There are several types of USING files:

1. INI.TLK files have already been discussed and are most often used to redefine the keyboard characters that control the operation of VersiCOMM.
2. INIARG.TLK files have also been discussed and are used by applications that chain to VersiCOMM to set flags, arguments and other parameters.
3. TLK files are files such as ELINK.TLK (below) that are most often used to dial and then logon to a service.
4. OFF files are normally used to automatically log off a service before disconnecting.
5. A trmdef.ATx file is used to "get the modem's attention," make sure it is active and properly configured for communications.
6. A trmdef.DLx file is used to dial the modem and to connect to a remote host.
7. A trmdef.HUx file is used to hang-up your modem.

The "trmdef" referred to above is the AMOS TRMDEF for the modem, such as MODEM1. The "x" in ".ATx", ".DLx" and ".HUx" is a character, such as "0" or "1", used to

differentiate sub-classes of a particular module type. For example, MODEM1.AT0 is the attention module for the port MODEM1 and MODEM1.AT1 is a variant attention module for the port MODEM1 that invokes MNP error correction.

## Using File Format

Take a look at the example USING file below. Note that, generally, there are three types of lines in the file. Lines preceded with "\$", lines preceded with ";" and lines not preceded with either of these characters.

The "dollar-sign" (ASCII 36) indicates that the remainder of the line following is to be interpreted as a VersiCOMM command. It is redefinable by using the command "COMMAND ascii-code", as described in the COMMANDS chapter.

When it is the FIRST character in the line a "semi-colon" (ASCII 59) indicates that the line is a "label" or "comment" line. If the semi-colon is followed by a numeral between 1 and 65535 it can be used as the destination of a VersiCOMM 'GOTO' command, otherwise the line is interpreted as a comment. Comments may be added to most VersiCOMM command lines, with the exception of WAIT, PRINT, QUIT and other commands that would interpret the comment as part of the command line. This character is redefinable by using the LABELCHR command as described in the COMMANDS chapter.

*THE CHARACTERS COMPRISING ANY LINE THAT IS NOT PRECEDED BY A "\$" OR A ";" ARE TRANSMITTED TO THE MODEM OR REMOTE COMPUTER UPTO AND INCLUDING THE CARRIAGE-RETURN.*

*IF A "\$" IS EMBEDDED IN A LINE ALL THE CHARACTERS UP TO THE "\$" WILL BE TRANSMITTED TO THE MODEM OR REMOTE COMPUTER AND THEN THE REMAINDER OF THE LINE WILL BE INTERPRETED AS A VERSICOMM COMMAND.*



BECAUSE IT IS COMMONLY USED IN LOGON COMMAND LINES FOR CERTAIN KINDS OF COMPUTERS, IF A ";" IS EMBEDDED IN A LINE, IT AND ANY FOLLOWING CHARACTERS ARE TRANSMITTED. EMBEDDED SEMI-COLONS ARE NOT INTERPRETED AS INDICATING THE START OF A COMMENT. COMMENTS SHOULD NOT BE INCLUDED ON LINES OF TEXT THAT ARE BEING SENT TO THE MODEM OR REMOTE COMPUTER.

The following USING file (ELINK.TLK) is an example of file that could be used (if you have an account) with a AT compatible modem to access AT&T's EasyLink service by executing the command, "COMM USING ELINK" at AMOS command level. Of course such a command can also be executed via command files and from Menu systems such as the COMMCENTRE. Then accessing a service such as EasyLink is as easy as pressing a key.

NOTE: A more complete version of this file will be found in account [7,121] of the VersiCOMM release tape.

```

;ELINK.TLK
;USING File to Access EasyLink
;Copyright 1984 Valley Programming Service, Inc.
;
$DIALOG                ;Dialog on
;*****
;* CONFIGURE           *
;*****
$BAUD 2400              ;Baud rate to 2400
;Check if modem is connected and operational
;Be sure modem is set to respond with words
$ATTENTION
$TSTFLG 3 0
$GOTO 100              ;No response from modem
;
$TREMOTE               ;Remote translation/filtering on
$REMOTE 17,0           ;Filter ^Q
$REMOTE 19,0           ;Filter ^S
$TLOCAL               ;Local translation/filtering on
$LOCAL 127,8           ;Translate del to a backspace
$LINEFEED OFF         ;No linefeeds when sending files
;*****
;* INITIATE LINK      *
;*****
$DIAL 18000000000
$TSTFLG 3 0
$GOTO 120              ;Did not connect
$PAUSE 20              ;Let modem catch its breath
;*****
;* LOGON REMOTE       *
;*****
$WAITIME 10            ;Wait 10 seconds for ID?
$WAIT ID?
$GOTO 130              ;No logon from EasyLink
01 EIDxxxxxxxxxxxxxxxx
$TALKTO                ;Goto conversational mode
;
;*****
;* OTHER EXITS        *
;*****
;100
$PRINT ?No response from modem...check if connected
$QUIT $$$$            ;Quit without hang-up
;120
$PRINT ?Did not connect to EasyLink
$QUIT
;130
$PRINT ?Connected, but no response from EasyLink
$QUIT

```



APPLICATION NOTE  
USING DOLLAR-SIGN IN LOGON SEQUENCE

A \$ included in a USING file line as part of a string of characters that is intended to be sent to the remote computer as part of a logon or command sequence will result in VersiCOMM attempting to interpret the characters following the "\$" as a VersiCOMM command. For example, if you included the password line "ABCDE\$FGHIJ" as part of a USING file, the "ABCDE" would be sent to the remote computer, but "\$FGHIJ" would be processed as a VersiCOMM command.

There are a number of ways to resolve such a conflict. The easiest would be to use the WRITECR command:

```
$WRITECR ABCDE$FGHIJ
```

Alternatively, the COMMAND character could be redefined to a character (including control-characters) other than dollar-sign for this USING file.

## The WAIT Command

Most VersiCOMM commands can be executed under the control of a USING file. There are a few VersiCOMM commands that are normally used ONLY in a USING file. One such command is the \$WAIT command. This command is a key ingredient to most USING files. Its format is as follows:

```
$WAIT wait-string
```

The \$WAIT command indicates that the characters that follow, up to but not including the carriage-return and line-feed that ends the line, are "wait characters" and are to be received AS A GROUP from the remote computer before processing of the USING file is continued. ONE space should separate the \$WAIT command and the wait string in the command line. Additional leading spaces will be considered significant and included in the wait string.

The interval of time that VersiCOMM will wait for the specified string is time specified in the last \$WAITIME command that was processed by the USING file. If no \$WAITIME command has been executed the wait time defaults to about 40 seconds.

*IF VERSICOMM RECEIVES THE STRING IN THE ALLOTTED AMOUNT OF TIME THE NEXT LINE OF THE USING FILE IS SKIPPED. IF THE EXPECTED RESPONSE IS NOT RECEIVED THE LINE IMMEDIATELY FOLLOWING THE WAIT LINE IS PROCESSED.*

This provides a means for executing a conditional 'GOTO' command depending on whether an expected response is received. VersiCOMM supports forward branching using the GOTO command, and backward branching using the BACKTO command.



APPLICATION NOTE  
WAIT CHARACTER SEQUENCE

When defining a sequence of wait characters it is important to note that receipt of "AxxBxxxxCDxxE" (x is any character) will NOT satisfy the wait string "ABCDE".





#### APPLICATION NOTE TIMING IN USING FILES

When first testing VersiCOMM's USING file processor, we discovered a curious characteristic of the Hayes Smartmodem. If upon the receipt of the "CONNECT" message from the Hayes, a string of characters is IMMEDIATELY sent to the modem, at least the first character of the string is likely to be lost. Due to the extremely small interval of time between the USING file receiving the "CONNECT" and then say, sending a password, the modem was not quite yet actually in the online state. This "gotcha" could occur with any "smart" type modem.

WHEN FIRST WRITING A USING FILE DIALOG, BE CONSERVATIVE, MANY DIALOG PROBLEMS CAN BE AVOIDED BY PAUSING A SECOND OR TWO AFTER A MODEM HAS ACKNOWLEDGED A PARTICULAR CONDITION. AFTER THE USING FILE IS WORKING YOU CAN GO BACK AND TRIM PAUSE TIMES IF IT IS NECESSARY. *THE DIALOG, EVEN IF IT APPEARS SIMPLE, SHOULD BE TESTED STEP BY STEP.*

Another timing problem that can occur is that the host computer can "detect" your modem's carrier and start to transmit data before your modem is ready to receive data. The result is that you miss all, or part of, the host logon message. This can occur with "MNP" type modems set for "auto-reliable" operation.

## Using File Operation - The WASIT Command


The \$WASIT command works just like the \$WAIT command, except that instead of waiting for new characters from the remote source it "replays" the first 512 characters received while the last \$WAIT command was active.

In effect this allows a USING file to act on any of several responses that could be received from a host. For example, the following script file segment waits for the string "38400", and if it is not received within 2 seconds, it checks to see if "19200" or "9600" was received instead.

```
$WAITIME 2
;
$WAIT 38400
$GOTO 1010
$TYPE 38400
$GOTO 2000
;1010
$WASIT 19200
$GOTO 1020
$TYPE 19200
$GOTO 2000
;1020
$WASIT 9600
$GOTO 1030
$TYPE 9600
$GOTO 2000
;1030
```

## Using File Operation - The CTRLC Command

The CTRLC command checks to see if a Control-C abort is pending and if so, branches forward (only) to the label line with the label-number specified.

When VersiCOMM is executing a script file, a  can have differing results depending on whether or not a WAIT command is being executed.

When a **Ctrl C** is keyed during a WAIT, but before the WAIT clears, VersiCOMM aborts the WAIT and executes the next line of the script file, just as if the WAIT had timed out. It also sets a flag that indicates a **Ctrl C** was keyed. Any following CTRLC command will detect that a **Ctrl C** is pending.

When a **Ctrl C** is keyed and a WAIT is not being processed, VersiCOMM aborts the execution of the script file and returns to Conversational mode. Any following CTRLC command is not effective, because the script has been aborted.



The CCENDTLK OFF command forces COMM to continue to execute the script file when a **Ctrl C** is keyed, and a WAIT is NOT being processed. A flag is also set so that any following CTRLC command detects that a **Ctrl C** has been keyed. The default state of CCENDTLK is ON.

The AMOS operating system Control-C abort condition is cleared by CTRLC.

## Using File Operation - The DIALOG Command

When testing a USING file it is essential that you be able to follow the processing of a USING file step by step. The DIALOG command can be used to turn the display of the USING file dialog ON or OFF. Following is an example of such a dialog. Responses from the modem are in bold. Responses from the remote computer are in italics. Notes concerning operation are in parentheses.

```


Dialing primary MCI number
Sending                               (send phone number to modem)
*WAITING FOR "CONNECT"

CONNECT
*WAIT CLEARED*
*WAITING FOR "name:"

Port: 7.
Please enter your user name:
*WAIT CLEARED*Sending                (send user name to service)
*WAITING FOR "word:"
USERNAME                             (service echoes user name)
Password:
*WAIT CLEARED*Sending                (send password to service)
End of USING file...
Entering conversational mode...       (we're talking to the service)
```

When DIALOG is ON and VersiCOMM starts to process a USING file the message "Processing USING file..." is displayed. During processing the following will be displayed:

- Any character received at the communications port is displayed. *This would include characters echoed or generated by the modem or remote computer.* Passwords and other characters sent TO the computer are not displayed unless they are echoed by the host. In the example above the string "USERNAME" was transmitted to the host IMMEDIATELY upon the receipt of the string "name:". The string was not echoed by the host until VersiCOMM was already waiting for the string "word:".
- When a WAIT command is encountered the message \*WAITING FOR "string" is displayed, where "string" is the string VersiCOMM is waiting for.

- When the string of characters specified in a WAIT command have been received from the host, the message "\*WAIT CLEARED\*" is displayed.
- If an expected "wait string" is not received from the host during the default wait time interval, or the time specified in a WAITIME command, the message "\*WAIT TIMED OUT\*" will be displayed.
- If a  is keyed while a WAIT is in effect, the WAIT will be aborted and the message "\*WAIT ABORTED\*" will be displayed.

After the processing of a USING file has been completed the message "END of USING file" will be displayed.

The state of the DIALOG switch can be tested in a script file using the TSTDIALOG command.

## The LISTEN Command

The function of a USING file is to operate VersiCOMM in the same way as a person would operate it. In addition to providing easy access to services, USING files can be used to automate an entire communications session, calling the service, logging on, issuing the service a series of commands, saving the result, disconnecting from the service, hanging up the modem and exiting VersiCOMM. The LISTEN command is used in such a USING file to put VersiCOMM in Conversational mode so that displayed data can be saved in a file. LISTEN will remain in Conversational mode as long as the host continues to display data. When the display of data has been completed, the LISTEN command returns control to the USING file so processing of the file can continue. The LISTEN command accepts an optional argument which is the number of seconds that LISTEN will wait, *with no data being displayed*, before returning control to the USING file. The default value of this parameter is about 30 seconds.

The following USING file segment illustrates the way in which the LISTEN command can be used in the retrieval of information in an automated session. It retrieves the a news summary from AT&T's FYI information service.

```
$WAITIME 30           ;Wait 30 seconds for ID?
$WAIT ID?
$GOTO 130             ;No logon from EasyLink
01 EIDxxxxxxxxxx
$WAIT PTS
$GOTO 140
/EXIT
$WAIT (RETURN)
$GOTO 150
$PAUSE 10
2
$WAIT CATEGORY NAME
$GOTO 160
$SAVE FYINWS.TXT
$PAUSE 10
NEWS
$LISTEN 10
$CLOSE
$QUIT
```

## USING File Termination

VersiCOMM will continue to process a USING file until one of four conditions is encountered:

1. If a TALKTO command is processed, control of VersiCOMM by the USING file is released and the user is put in Conversational mode.
2. If the end of the USING file is reached, VersiCOMM proceeds as if a TALKTO command had been processed.
3. When a QUIT command is processed, VersiCOMM checks for an ".OFF" logoff file and ".HUx" hang-up file, processes them if found, and then terminates.
4. An EXIT command causes VersiCOMM to terminate without the normal checks for ".OFF" and ".HUx" files.

## OFF Files

An OFF file is a special kind of talk file used to save a particular host's logoff sequence and automatically log you off from that service when you exit VersiCOMM. If a USING file was processed when COMM was first executed, VersiCOMM checks if there is an .OFF file with the same name (and in the same account) as that first USING file. If it finds one it processes it. For example, if you use DSK0:MCI.TLK[7,121] to access MCI Mail, DSK0:MCI.OFF[7,121] will log you off when you exit VersiCOMM.

This feature provides a "service independent" way of disconnecting from an arbitrary service.

## HANGUP File

After processing any OFF file, VersiCOMM next looks to see if the file trmdef.HU0 (or HANGUP.TLK) exists in the account of the first USING file processed, where "trmdef" is the AMOS TRMDEF name of the modem, e.g. "MODEM1". If no USING file was processed it looks in the current user account. If trmdef.HU0 (or HANGUP.TLK) is found it is processed as a USING file and sent to the host before exiting. For example, the following file is a simplified version of the file included with VersiCOMM that is used to hang-up a AT compatible smart modem with the escape character set to "+++".

```

;MODEMX.HU0
;USING File to hang-up AT compatible modem
;Copyright 1984 Valley Programming Service, Inc.
;
$DIALOG    ON
$WAITIME 3      ;Wait 3 seconds for modem responses
$PAUSE 12       ;Pause 1.2 seconds
++$PAUSE 12     ;Send "+++" NO CARRIAGE RETURN and pause again
$WAIT OK

$PAUSE 1        ;Let modem catch its breath
ATH0
$WAIT OK
$PRINT **!! Modem may not have hungup properly !!**

```



BE CAREFUL NOT TO INCLUDE A "QUIT" COMMAND IN A HANGUP FILE, SINCE AN ENDLESS LOOP WOULD RESULT.

If you often use TLK files that set PARITY and SIZE to non-standard values it may be desirable to add PARITY and SIZE commands to the HANGUP TLK file to reset these parameters to the standard values.



APPLICATION NOTE  
VPS ALPHABASIC COMMUNICATION SUBROUTINES

During the development of the USING file concept it became clear that if we were not careful we would re-invent the BASIC language to control the logical flow in USING files. Our research indicated that a simple, sequential logic combined with a conditional branch based on the WAIT command would handle a substantial majority of the access requirements of information and communication services.

If you have a very demanding application that involves a complicated interactive dialog, with many different decisions that must be made based on the varying responses of a remote computer, VersiCOMM-Plus includes an interface and subroutines that enable the use of AlphaBASIC as a communications processing language. VPS also has available a set of Communication Subroutines that can be used independently of VersiCOMM.



## Sending and Receiving Files

---

### Transmitting Files

The transmission of files to another system is a basic function of most communications programs. VersiCOMM can send files in several ways. Which method is employed depends on the features and capabilities of the host system. Following is a list of the commands that can be used to transmit a file using VersiCOMM-Plus:

SEND	Send a sequential file without protocol.
SENDEPX	Send a sequential file using echoplex error checking.
SENDL	Send a sequential file without protocol a line at a time.
SENDTLK	Execute a VersiCOMM script file
TRANS	Send AMOS file(s) to another Alpha Micro using a MNP-based, proprietary protocol.
SENDX	Send a sequential file using XMODEM or XMODEM 1K protocol.
SX	A version of SENDX adapted for remote use.
SENDZ	Send sequential file(s) using the ZMODEM protocol.
SENDZ/REMOTE	SENDZ with switch for remote use.
SENDZ/YMODEM	Send sequential file(s) using the YMODEM protocol.
BPXFER	Send sequential file(s) using Compuserve B Plus protocol.

### SEND

SEND is most frequently used to transfer a file to a remote system that is not an Alpha Micro and does not support XMODEM, YMODEM, ZMODEM or the B Plus protocols. The general requirements for being able to SEND a file to another system are:

1. The file must be a sequential file.
2. The remote system must have some facility for receiving a file "without protocol."
3. The remote system must be capable of RS-232 communication.

Nearly all systems capable of asynchronous communications have some facility for receiving a file "without protocol." If no other facility exists a text editor can sometimes be used for this purpose.


The program MINREC, included with VersiCOMM, is a very simple example of a receive program. If MINREC is resident on a remote Alpha Micro, it can be used to receive a sequential file transmitted by SEND.

To meet special system requirements the operation of SEND can be modified in several ways.

### **Speed Control**

On AMOS/L systems the transmission speed of send can be "throttled" down from the nominal baud rate to facilitate transfer to a system overburdened with input, or for some other reason unable to handle transmission at full speed. This feature should not be confused with changing the actual baud rate.

If desired you can specify in the SEND command line the percentage amount you wish to decrease the transmission speed. For example,

```
SEND FILE.LST 50 
```

would decrease the transmission speed by approximately 50%. Note that a 100% reduction would double the transfer time. The maximum reduction that may be specified is 65535%

### **End-of-Line Delay**

An end-of-line delay can be specified using the EOLTIME command. SEND will pause for the time specified when the currently defined end-of-line character is detected. The end-of-line character is defined using the EOLCHR command.

### **Flow Control**

Some remote systems have the capability to detect when they have received too many characters and send back a special character that tells the transmitting system to STOP!! Then, when ready to accept input again, the remote system sends another character that means GO AHEAD!! This capability is called "flow control," and the dialog involved is sometimes referred to as "handshaking."

XON/XOFF flow control uses a Control-S to suspend transmission, and a Control-Q to start it up again. When the HANDSHAKE switch is set ON, SEND will respond to the XON/XOFF flow control characters.

If a remote system uses two other characters for this purpose, VersiCOMM's translation facility may be used to convert whatever the remote system uses for XOFF to Control-S, and XON to Control-Q.



## Linefeed Control

It is common for computers to add a linefeed to any carriage return received in some states of operation but not in others. A receive program written in AlphaBASIC would be expected to add a linefeed, while an assembly program such as MINREC may not. The easiest way to determine what a particular receive program will do is to send it a short text file with LINEFEED set ON, and then read the file on the remote system to see if it contains unnecessary linefeeds.

## Translation and Filtering

If TLOCAL is set ON, any translation or filtering specified by previous LOCAL commands will be effective AS THE FILE IS TRANSFERRED. For example, if a LOCAL 65,66 had been executed, any "A" in the transmitted file would be translated to a "B" before sending it on to the remote computer.


## SENDEPX

SENDEPX sends a file with "echoplex" error detection.

While most communication services now support access with MNP error correction and file transfers with a standard protocol, from time to time it may still be desirable to send a file with echoplex error correction. SENDEPX provides a means for detecting errors in file transfers to systems that echo characters as they are received and stored. If SENDEPX detects an error its prompt changes from a hyphen to an asterisk. If an error was detected during the transfer the message "%Possible error detected during file transfer%" will be displayed, and SENDEPX will remain in Command mode instead of returning to Conversational mode.

SENDEPX should be used when an error-correcting protocol is not available. It verifies that all transmitted characters are echoed back correctly. To go undetected a character that was corrupted by a transmission error would have to be corrupted on the echo to the character originally transmitted. The probability of this occurring is about 1 in 16000.

The rate that SENDEPX can send a file is dependent upon the rate that the remote system can echo characters. SENDEPX sends a packet of 64 characters and checks the character stream as it is echoed. In order to provide some control over the rate of the transfer smaller packet sizes may be selected by specifying an argument following the filename in the SENDEPX command line. For example,

```
SENDEPX FILE.LST 10 
```

would use a packet size of ten characters.

When using SENDEPX it is important that it be used to send only the actual text of the message, and not any additional commands that would result in characters other than those transmitted. SENDEPX will ignore the extra linefeeds that result from sending carriage returns and any nulls that the service may use for carriage return or linefeed padding.

The HANDSHAKE switch has no affect on the operation of SENDEPX. No local translation or filtering is effective when using SENDEPX.

## SENDL

The SENDL command is used for sending text to a line editor that prompts the user for each line of text. For example:

```
1:This is the first line of text.  
2:
```

Before sending the file the prompt character must be defined using the PROMPTCHR command and the end-of-line character must be defined using the EOLCHR command. The default prompt character is a colon. If PROMPTCHR is set to a null (ASCII 0), SENDL does not wait for a prompt before sending the next line. (The end-of-line time should normally be zero when sending text in this manner.)

Linefeeds will be suppressed or sent, depending on the state of the LINEFEED switch. Normally you would suppress linefeeds when using SENDL.

*TRANSLATION OF CHARACTERS WILL BE PERFORMED IF LOCAL TRANSLATION IS ENABLED.*

An optional throttle percentage may be specified in order to slow the speed of the transfer by the percentage indicated. In the command example a throttle percentage of 50% has been specified. The maximum percentage that may be specified is 65535%.

## SENDTLK

SENDTLK is normally used to initiate the processing of a USING file in the middle of a communications session. If there were a standard series of commands you would issue a service in a particular situation, the sequence could be saved in a USING file and sent to the service using the SENDTLK command.

## TRANS

TRANS is a VersiCOMM-Plus program that will transmit any type of AMOS file to a remote Alpha Micro with CRC16 error detection and correction. Throughput to small or heavily burdened systems can be significantly increased by using **FastQ**, our enhanced serial communications system.

TRANS increases the effective speed that data is transferred by using either **FasTrans** Data Compression or Adaptive Data Compression. As a file is transferred TRANS analyzes the content of the file "on the fly" and then reassigns short bit patterns to represent frequently used characters or strings of characters and long bit patterns to represent infrequently used characters. Virtually all (non-compressed) files are compressible, not just files with many repeated characters. For example, data such as this text would be compressed 40%-50% using Adaptive Compression and even a higher percentage with **FasTrans** compression.

TRANS requires the remote computer have resident a companion module, RZVA, to handle the remote end of the transfer. RZVA must be executable from the account in which you are logged on the remote computer.



THE PROGRAM <b>RZVA</b> MUST NOT BE RENAMED!
--

RZVA uses the **FastQ** enhanced serial communications system if the file RZVA.INI exists (on the remote system) in the account from which RZVA is being run, or if RZVA.INI exists

in DSK0:[7,0]. The single line "FASTQ ON" should be included in this file. If the file RZVA.INI is not found, *FastQ* is not used.

Once a link has been established using VersiCOMM, TRANS can be run at AMOS command level or directly from VersiCOMM.

TRANS accepts wildcards and output filespecs similar in operation to the AMOS COPY command. It can transfer all types of AMOS files including text and binary sequential files, and contiguous (random) files. If needed, TRANS will create an account on a remote system if the user is logged into the operator account of the remote system.

The MEM: device may be specified in input or output file specifications. If a remote system has been booted under special circumstances, such as a warm boot, software may be transferred to remote user memory if RZVA.LIT and MEM.DVR are already in memory or can be loaded from disk.

TRANS automatically configures itself for optimal reliability when transferring files to a "slow" remote system, such as an AM1000, at 9600 bps and higher baud rates. It also automatically adapts to the condition of the line or the burden of the receiving computer by adjusting its data rate and packet size.

In addition to the usual link level packet-oriented CRC error detection, an application level CRC hash total is calculated on the entire file and transmitted at the end of the file. If this hash total does not match that of the received file, the file is rejected. This mechanism protects against protocol and other application level software errors.


If requested, TRANS will maintain a LOG file that records the date and time of the transfer, the name of the file transferred, the total number of files transferred during the session, the total number of retries and the average transfer rate.

TRANS does not require immediate acknowledgment of transferred packets. This increases throughput on satellite links, or other links with line turnaround delays. The number of outstanding packets can be adjusted from as few as one to as many as eight packets.

TRANS does not expend the CPU time of the remote system unnecessarily. CPU time is conserved unless the transfer is at 9600 bps or a higher rate, and the receiving system is an older system such as an AM1000.

## Operation

TRANS may be run from AMOS command level, or directly from VersiCOMM. The general form of the TRANS command is:

```
TRANS {remote-fspec=} local-fspec {(terminal-name)}{/switches} 
```

Type TRANS followed by the name of the file you wish to transfer followed by the name of the terminal, enclosed in parentheses, associated with your modem or other RS-232 link to the other computer.

Since ersatz definitions may differ from system to system, ersatz device specifications should not be used in the remote-fspec.

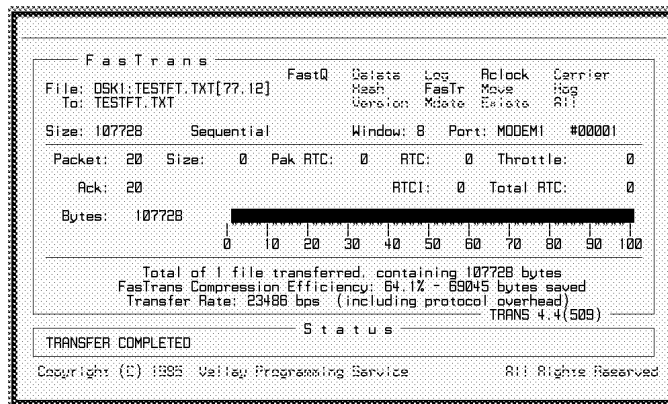
If you do not specify a terminal-name TRANS uses MODEM1, unless you are running TRANS from the VersiCOMM-Plus command prompt, in which case TRANS will use the modem that VersiCOMM is already using.

Following are examples of TRANS command lines with and without a terminal-name specified:

```
TRANS FILE.TXT
```

```
TRANS FILE.TXT(MODEM3)/D
```

The program will then respond with a display similar to the following:



The display is updated as the file is transferred.

The upper section of the window displays the context of the file being transferred. The following information is provided:

File	File specification of local file.
To	File specification of remote file.
Size	Size of file transmitted in bytes.
Sequential	Type of file. "Random" is an alternate type of file.
Window	- Is the number of packets that can be outstanding without an ack.
Port	The terminal definition (TRMDEF) of the communications port that is being used by TRANS.
#00001	A running total of the number of files transmitted.

The middle section of the window displays the current state of the transfer, including:

Packet	Sequence number ranging from 0 to 255 of the last packet transmitted.
Ack	Last packet acknowledged.

Size	Number of data bytes in the packet.
Pak RTC	Retry count for the current packet.
RTC	Retry count for the current file.
Total RTC	Cumulative retry count for all files.
RTCI	Retry count index.
Throttle	Percentage reduction in speed of transmission.
Bytes	Number of data bytes input from the file for transmission.


The middle section of the window also includes a bar graph that indicates the percentage of the file that has been transferred.

The lower sections of the display lists status messages and, upon completion of the transfer, the following statistics:

- The number of files transferred.
- The total number of bytes in the files transferred.
- The percentage compression achieved for all files transferred.
- The number of bytes that did not need to be transferred as a result of data compression.
- The average transfer rate.


The average transfer rate is calculated by dividing the total bytes in the files transferred by the total time for the transfer. The actual data rate is higher because of the additional bytes that must be transmitted as part of the file transfer protocol.

TRANS supports wildcarding and other features supported by the AMOS system file, CMDLIN.SYS. Multiple files of differing types may be specified and sent with a single command. In general conventions followed by the AMOS command COPY apply to TRANS. For example, if you were logged in the OPR: account on both the local and remote system the following command,

```
TRANS *.NEW[300,1]=*.BAS[200,70] 
```

would transmit all of the BAS files in [200,70] of the local system to [300,1] of the remote system, renaming them to NEW files and creating the account [300,1] if necessary.

Files can be read from several different accounts on the local system and sent to one account on the remote system. For example,

```
TRANS =*.BAS[200,50],*.BSI[200,0] 
```

would transfer the files specified to whatever account the user is logged into on the remote system. While,

```
TRANS *.BAS[200,50],*.BSI[200,0] ❷
```

would attempt to transfer the BAS files to [200,50], and the BSI files to [200,0] of the remote system. This would require that you be logged into one of the [200,x] accounts on the remote system.

With the implementation of CMDLIN.SYS wildcarding the placement of switches in the TRANS command line is much more flexible than in earlier versions of TRANS. When using TRANS from AMOS command level, switches may be placed anywhere in the command line.

In addition to wildcarding, CMDLIN.SYS supports the specification of files via an "indirect file." That is, the files to be transferred by TRANS can be specified in a file. For example, if the file "FILES.IND" contained the following lines:

```
FILE1.DAT,  
FILE2.DAT,  
FILE3.DAT/D
```

The command TRANS @FILES would be equivalent to TRANS FILE1.DAT,FILE2.DAT,FILE3.DAT/D. Note that even though the file specifications are on separate lines of the file, they are appended to form one file specification.

### File List Processing

File list processing using the /FILELIST switch is not subject to the same limitations as CMDLIN.SYS Indirect files. A FILELIST file can be as large as the operating system will allow..

The FILELIST switch is used in the following manner,

```
TRANS/FILE:listfile *.* ❷
```

where "listfile" is file, such as a VERIFY directory file, that lists each file to be transferred. The files must be listed one file specification per line, with the file specification located at the beginning of the line. Additional information, such as version information, or a hash total, can follow the file specification as long as it is separated from the file specification by at least one space.

Files in the list that are not found or are incorrectly specified are skipped and the next file in the sequence processed.

### Option Switches

/FASTRANS or /NOFASTRANS

If adequate memory is available, TRANS uses *FasTrans* data compression as a default. This type of compression normally reduces the number of characters transferred by 45% to 75% or more. It compresses virtually all (uncompressed) files. A text file such as this would be compressed by about 65%!

A fixed /PACKET size of 9000 is used when *FasTrans* compression is specified. A smaller packet may be used, but this will reduce throughput.

If insufficient memory is available to use *FasTrans* compression, then Adaptive compression is used. To force the use of Adaptive compression use the switch /NOFAST.

/ADAPT or /NOADAPT

TRANS will fallback to Adaptive compression when *FasTrans* compression cannot be used. This type of compression reduces the number of characters transferred by 25% to 50% or more. It compresses virtually all (uncompressed) files. A text file such as this would be compressed by about 40%.


If the link speed and receiving computer burden permit, a packet size of about 500 may improve throughput by another 10% or so. See the discussion of the /PACKET and /MAXIMUM switches below.

If the system running TRANS is slow or heavily burdened Adaptive compression may be a better choice than *FasTrans* compression, since it is less CPU intensive.

When /NOADAPT is specified, TRANS reverts to "run length" compression, which is much less CPU intensive, but efficient only on those files that contain many repeated characters, such as nulls in a partially full random file.

/FILELIST or /NOFILELIST

In addition to using wildcarding to specify a group of files to be transferred, TRANS can process a file listing, such as a VERIFY list file. The syntax for using the FILELIST switch is:

```
TRANS/FILE:listfile *.* 
```

The wildcard file specification "\*.\*)" must be included in the command line. See the section "File List Processing" for additional information.

/LOG or /NOLOG

If specified, TRANS will log the name of the file transferred and the number of retries that occurred during the transfer, in the file TRANS.LOG. If the log file already exists new entries will be appended. The entire filespec, including device and ppn, is listed in the log file entry. If the /INFO switch is set (the default), the effective rate of the transfer will also be logged.

/RETRY:n

Controls the number of times TRANS will attempt to establish a remote link (n/2 times), resend packets (n times) and how many consecutive retransmissions of different packets may occur without a valid acknowledgment (n packets). If /R is specified but n is not specified, 50 is used. If /R is not specified, 12 is used.

/DELETE or /NODELETE

The /DELETE switch indicates that a file should be transferred to the remote system, even if it already exists on the remote system. When this switch is specified, if the file does exist on the remote system, it is renamed to have an extension of ".FTB" PRIOR to the actual transfer. This is important to remember if the transfer is interrupted!

/UPDATE or /NOUPDATE also /HASH or /NOHASH

The /UPDATE or /HASH switch indicates that a file should be transferred to the remote system if it does not exist on the remote system, or if it does not have the same hash total as the file on the remote system. If the file on the remote system has the same hash total, the local file is not transferred. If the local file is transferred to the remote system, the original remote file is renamed with an extension of ".FTB" PRIOR to the transfer. The /UPDATE and /HASH switches are functionally identical.

/VERSION or /NOVERSION

This switch indicates that a file should be transferred to the remote system if it does not exist on the remote system, or if the (local) file has a later version number than the file on the remote system. If the file on the remote system has the same or later version number, the local file is not transferred. If the local file is transferred to the remote system, the original remote file is renamed with an extension of ".FTB" PRIOR to the transfer.

/MDATE or /NOMDATE

The /MDATE switch indicates that a file should be transferred to the remote system if it does not exist on the remote system, or if the (local) file has a 'modified' date later than the file on the remote system. **NOTE: BOTH THE LOCAL AND THE REMOTE FILE MUST BE DATE STAMPED FOR THE TRANSFER TO TAKE PLACE.** If the local file is transferred to the remote system, the original remote file is renamed with an extension of ".FTB" PRIOR to the transfer.

/EXISTS or /NOEXISTS

This switch indicates that a file should be transferred to the remote system **ONLY** if it exists on the remote system. It may be used in combination with other switches such as /HASH, /VERSION, and /MDATE.

/RANDOM or /NORANDOM

The /RANDOM switch indicates that **ONLY** random files in the wildcard set are to be transferred. Note that /NORANDOM means not "only random files."

/SEQUENTIAL or /NOSEQUENTIAL

The /SEQUENTIAL switch indicates that **ONLY** sequential files in the wildcard set are to be transferred. Note that /NOSEQUENTIAL means not "only sequential files."

/MOVE or /NOMOVE

When this switch is specified, the transmitted file (the local file) is renamed to have an extension of ".FTB" when the transfer has been completed.



/WINDOW:n

TRANS normally does not require immediate acknowledgment of transmitted packets. Up to 8 packets may be outstanding. In most situations this increases throughput, especially on links that have delays, such as satellite links. The

/W switch enables the user to specify the number of packets that may be outstanding, ranging from 1 to 8. Using /W:1 can help when the default of /W:8 overruns the remote system. It can also be helpful if it is necessary to use a small packet size when transferring files via a DATA RACE or similar modem. (See the discussion of /PACKET below.)

/HOG or NOHOG

When RCLOCK is installed and running in SYSTEM memory on the receiving system, this switch forces the remote receiver into a high-speed "hog the CPU" mode. The receiver automatically uses this mode if the receiving system is a 68000 or 68010 and the modem port is set for 9600 bps or faster. This switch has no effect if RCLOCK is not installed.

/PACKET:n and /MAXIMUM:n

The /P:n and the /M:n switches have similar functions. The /P switch permits the user to specify a FIXED packet length, while the /M switch is used to specify a maximum VARIABLE packet length. A packet size of up to 9000 bytes may be specified. When Adaptive compression is selected, TRANS uses a variable packet size based on the number of retries that have recently occurred. When *FasTrans* compression is selected, a fixed packet size is used.

When using DATA RACE modems or similar modems with high and low speed channels, use of the /P switch prevents acknowledgments from choking the slow channel, due to many small packets being transmitted on the high speed channel. Use a fixed packet size of about 400.

Use of a large packet size decreases the protocol overhead, but increases the input overhead on the receiving system. If the receiving system is slow or heavily burdened, *FastQ* Enhanced Input processing can increase performance dramatically.

When the receiving computer is very burdened, or the line very noisy, the /SPRINT switch will often get a file through faster than more conventional configurations. A very small packet size may be required in unusual circumstances.

/SPRINT

This is a convenience switch that is equivalent to:

/WINDOW:1/PACKET:100/HOG

Use this configuration when transmitting a file at high speed to a slow or burdened system.

/KILL

The /K switch is used to abort a remote RZVA that has not timed out. For example, if TRANS terminates due to noisy line conditions, RZVA will be left on its own and will time out in about two minutes. Normally you would disconnect and then reestablish the connection with the hope of getting a better line. Upon re-establishment of the connection if the remote does not appear to be responding, RZVA has probably not timed out. Entering the command "TRANS/K" will abort RZVA.

NOTE: If the remote RZVA is up to date, TRANS can transmit to RZVA before it has timed out.

#### /CARRIER

When this switch is specified, a change in the state of DCD (usually pin 8) from the state detected when TRANS starts results in TRANS aborting the transfer.

#### /INFO or /NOINFO

This switch results in the display of information about the amount of compression achieved (when using adaptive compression) and the actual throughput of the transmission, including wildcarding and protocol overhead. Note that throughput info may not be accurate on very short transfers.

#### /EXCLUSIVE or /NOEXCLUSIVE

This switch determines whether the file on the transmitting system is opened for exclusive or shared access.

All switches are global. They affect all files regardless of where they are placed in the command line.

When FasTrans compression is used the default configuration is:

```
/FASTRANS/NOFILELIST/NOLOG/RETRY:12/NODELETE/NOUPDATE/WINDOW:8
/PACKET:9000/NOSPRINT/NOMOVE/NOMAXIMUM/NOKILL/NOCARRIER/INFO
/NOVERSION/NOMDATE/NOEXISTS/NORANDOM/NOSEQUENTIAL/EXCLUSIVE.
```

When Adaptive compression is used the default configuration is:

```
/ADAPT/NOFILELIST/NOLOG/RETRY:12/NODELETE/NOUPDATE/WINDOW:8
/NOPACKET/NOSPRINT/NOMOVE/MAXIMUM:200/NOKILL/NOCARRIER/INFO
/NOVERSION/NOMDATE/NOEXISTS/NORANDOM/NOSEQUENTIAL/EXCLUSIVE.
```

If /NOADAPT is specified /MAXIMUM:512 is used.

### **Troubleshooting**

#### **Cannot work interactively on the remote Alpha Micro.**

*Problem is not related to file transfer.*

**Can work interactively on the remote Alpha Micro, but cannot initiate file transfer. TRANS gives the message, "?Cannot establish remote link."**

*Check that TRANS is using the same port as that being used when working interactively on the remote system.*

*Check that an executable copy of RZVA exists on the remote system. Normally the file RZVA.LIT should reside in DSK0:[1,4] or system memory.*

*Be sure that the communications port and any intermediate device such as a modem is set for 8 data bits. Normally all ports on the Alpha Micro are set to 8 data bits when the system is booted. The comm port is set to 8 data bits when COMM is first executed, but can be changed to other configurations by using the SIZE, PARITY and STOP commands.*

**The file transfer proceeds normally for part of the transfer and then abruptly the retry count increases until finally TRANS gives the message,**

**"Retry count exceeded--no valid remote responses."**

*Check if the communication link has been interrupted.*

*Be sure that XON/XOFF flow control has been disabled and that XON/XOFF pass-through has been enabled on the modem or any other intermediate device in the communications link. Some 3rd party interfaces also support XON/XOFF flow control at the port level. Be sure this is disabled.*

**When the file transfer is initiated, TRANS outputs the message "%Adaptive compression not supported on remote."**

*This message results when the RZVA.LIT program on the remote system is a version that does not support Adaptive Data Compression. Often this occurs when RZVA is inadvertently replaced when the operating system is updated. RZVA.LIT is included as part of the operating system, but usually is an earlier version than that being distributed with VersiCOMM.*

## **Error Messages**

### **%Adaptive compression not supported on remote**

The remote server RZVA.LIT is not current and does not support Adaptive compression.

### **?Assigned FAX job not running FAX program**

The FAX job was not running FAXSER or other FAXxxx program.

### **?Cannot output to OPR:BADBLK.SYS on remote system**

TRANS cannot transfer a file to OPR:BADBLK.SYS.

### **?Cannot transfer an 'empty' sequential file**

The file specified is empty, i.e. it contains no data bytes.

**?Communications port is in use**

The terminal selected for use in communications (usually a modem) is already in use by another program.

**?Could not add PPN on remote system**

In order to accomplish the transfer it would be necessary to add a new PPN on the remote system, and an error occurred when this was attempted. Usually this is the result of a "protection violation" on the remote system.

**?Could not establish remote link**

The remote computer either did not respond to repeated commands to initiate the link. Either quality of the communications line was very poor or RZVA was not executable on the remote computer.

**?Could not find block to retransmit**

Internal error...notify Valley Programming Service if it occurs.

**?Could not find PSEUDO driver**

TRANS uses the PSEUDO driver for communications during file transfers. This driver is part of the monitor and should always be found.

**?CRC sum of local file and remote file did not match**

The hash total of the file transmitted does not match that of the file received. Notify Valley Programming Service of ANY occurrence of this message.

**?Error opening file on local system**

TRANS could not OPEN the file specified on the local system due to a protection violation or other system error.

**?Error opening file on remote system**

TRANS could not OPEN the file specified on the remote system due to a protection violation or other system error.

**?Ersatz device is not allowed in output filespec**

Since an ersatz device may not have the same definition on a local and remote system, ersatz device specifications are not valid in an output filespec.

**?Error in wildcard file processing**

An error was reported by the CMDLIN.SYS wildcard file processing system.

**%FasTrans compression not supported on remote**

The remote server RZVA.LIT is not current and does not support *FasTrans* compression.

**?FAXCOM not responding, FAXSER not restarted...**

FAXCOM did not respond when TRANS attempted to restart FAXSER.

**?FAX job not in CMD or COM state**

The FAX job was not running FAXSER, FAXCMD or FAXCOM.

**?File not resident on remote system--exists switch specified**

The /EXISTS switch was specified and the file is not being transferred because it does not exist on the remote system.

**?File resident on remote system--date is the same or later**

The file is not being transferred to the remote system because the /MDATE switch was specified and the date stamp on the remote file is the same or is later than the date stamp on the local file.

**?File resident on remote system--date not available**

The file is not being transferred to the remote system because the /MDATE switch was specified and the remote file has no date stamp.

**?File resident on remote system--delete switch not specified**

The delete switch /D must be specified in order to transmit a file which already exists in the account into which you are logged on the receiving computer.

**?File resident on remote system--hash codes are the same**

This message is output when the /UPDATE or /HASH switch has been specified and a file is skipped because the transfer of the file would be redundant.

**?File resident on remote system--version is the same or later**

This message is output when the /VERSION switch has been specified and a file is skipped because the transfer of the file would be redundant.

**?File in LIST not found**


A file specified in a FILELIST could not be located according to the file specification in the FILELIST file.

**?File in LIST not specified correctly**

A file in the FILELIST file was not correctly specified.

**?FILE switch requires a file specification**

The /FILELIST switch was used without including a file specification. The correct syntax is :

```
TRANS/FILE:filespec *.* 
```

**?Memory allocation error: FasTrans compression**

TRANS attempts to evaluate if adequate memory is available to use *FasTrans* compression. This message results if TRANS calculated it had enough memory, but in fact not enough memory was available.

**?MDATE switch specified--no date stamp on local file**

The /MDATE switch requires that the local file being transferred and the remote file, if it exists, have a date stamp.

**?Not enough memory for FasTrans compression**

This message is briefly displayed if TRANS calculates that there is insufficient memory to use *FasTrans* compression. When this occurs, TRANS falls back to Adaptive compression.

**?Not enough space available for file on remote system**

You tried to transfer a random file that would not fit on the receiving computer.

**?No files on the local system meet your criteria**

TRANS could not locate any file matching the file specification(s) in the TRANS command line, with the switches (if any) specified. This will be the last message displayed, and may overwrite a previous message, such as "?File resident on remote system--delete switch not specified."

**?OIP timeout**

The terminal's output in progress flag did not clear when expected.

**?Remote transmit program TNZA is active**

If the communications link is interrupted while RETRVing a file and then reestablished, this message may be displayed if you immediately attempt to TRANS a file.

**%Remote active -- attempting to restart...**

The remote server RZVA.LIT was already running when the TRANS command was executed. TRANS will attempt to perform the transfer requested.

**%Remote is 68000 or 68010 -- using WINDOW:1**

If the remote server RZVA reports it is running on a 68000 or 68010, the port baud rate is 9600 bps or greater and *FastQ* is not in use, TRANS defaults to a WINDOW of one. In some cases a larger WINDOW may still result in a faster transfer. If a WINDOW parameter is specified in the command line it will override this action.

**?Retry count exceeded--line too noisy**

TRANS tried to send the same block more than 12 times, or the currently defined maximum retry count.

**?Retry count exceeded--no valid remote responses**

TRANS received no valid acknowledgments while transmitting 12 consecutive blocks, or a number of blocks equal to the currently defined retry count. This error will usually result from a link failure, i.e. loss of carrier or one of the modems inadvertently hanging up.

**?Terminal not found**

The terminal specified, or the default terminal MODEM1, was not defined in your system as a terminal.

**?Transmit buffer overflow -- use NOADAPT switch**

When using Adaptive compression, TRANS assembles packets a bit at a time. Assembly ends when a break occurs on a byte boundary that equals or exceeds the currently specified packet size. This will normally occur within a few bytes of the desired packet size. It is unlikely, but possible, that packet assembly does not break at a byte boundary within the buffer size allocated.

Should this occur, specifying the /NOADAPT switch will change the method of compression to a method that is byte oriented, and will not overflow the transmit buffer.

**?Unknown error on remote**

An unexpected and unknown error occurred while transferring the file to the remote system.

**SENDX AND SX**

SENDX and SX are Xmodem compatible programs used to send sequential AMOS files to another asynchronously linked computer equipped with a Xmodem compatible software package. Both CRC16 or CHECKSUM modes of operation are supported.

Normally SENDX is used by an Alpha Micro user to send a file to a remote PC, and SX is used by a PC user to send a file from a remote Alpha Micro to the PC.

Once a link has been established using VersiCOMM, SENDX can be run at AMOS command level or directly from VersiCOMM.

**SENDX Operation**

The format of the SENDX command is:

```
.SENDX filespec (terminal-name)/switches
```

To use SENDX you would first use VersiCOMM to establish communications with a remote system that supports the XMODEM protocol. Next, on the remote system, you would enter the command(s) to receive a file on the remote system using the XMODEM protocol. This process is often referred to as "uploading a file."


After you have started the XMODEM receive on the remote system, switch to VersiCOMM command mode and type SENDX followed by the name of the file you wish to transfer. For example,

```
SENDX MOVCUR.TXT
```

If SENDX is being run from the AMOS command level it will assume the terminal-name for the modem port is MODEM1, unless an alternate terminal-name is specified in parentheses. For example,

```
SENDX MOVCUR.TXT (TERM3)
```


The program will then respond with a display of the status of the transmission similar to the following:

```
.SENDX TEST.M68   
Waiting for initial NAK from receiver...^C to ABORT  
NAK received -- will use checksum error detection  
Number of 128 byte blocks: 87  
Blocks sent: 87 Retries: 0  
End of file  
Final response from receiver:  
.
```

Note that the block size is 128 and not the usual 512 byte AMOS block.

## SX Operation

The format of the SX command is:

```
.SX filespec/switches 
```

To use SX, first use the communication program on your PC (or other computer) to establish communications with the Alpha Micro from which you wish to send a file. When you have logged onto the Alpha Micro and can enter commands, type SX followed by the name of the file you want to transfer. For example,

```
.SX MOVCUR.TXT 
```

SX will then output the message,

```
Waiting for initial NAK from receiver...^X^X to ABORT
```

At this point initiate the XMODEM receive process using the procedure prescribed for your PC's communication software.

## Switches

- /Z - Add a CPM EOF indicator to the end of file.
- /Y - Use YModem 1024 byte blocks.

## Errors

### ?File not found

SENDX (or SX) could not locate the file as it was specified.

### ?Terminal not found

The terminal specified, or the default terminal MODEM1, was not defined in your system as a terminal.

### ?Communications port is in use

The terminal selected for use in communications (usually a modem) is already in use by a communications program.

### ?Could not find PSEUDO driver

SENDX uses the PSEUDO driver for communications during file transfers. This driver is part of the monitor and should always be found.



**?Did not receive initial NAK**

The remote receive program did not tell SENDX (or SX) to start sending by sending it an ASCII "NAK" code.

**Transfer canceled by receiver**

The program was aborted using the control X command or was canceled after ten consecutive errors.

**?More than 10 retries -- transfer canceled**

The same block was transmitted 10 times unsuccessfully.

**SENDZ**

SENDZ transmits sequential files using the ZMODEM or YMODEM protocols. ZMODEM is generally regarded a much more robust and efficient protocol and if the sending system supports both protocols, ZMODEM should normally be used. ZMODEM was originally developed to alleviate throughput problems experienced by packet-switched network customers using the XMODEM and KERMIT file transfer protocols. It is the most capable of the file transfer protocols offered on AMTEC+.

Unlike XMODEM, it does not pad the last block of a transferred file. Much faster than KERMIT, ZMODEM uses a streaming protocol with very low overhead and a 32-bit CRC checksum. In addition, multiple files may be downloaded in a single transfer, and should the transfer be interrupted you can dial back into the system and resume the transfer at the point at which it was interrupted.

For SENDZ to properly execute, RCLOCK must be installed and SENDZ must be run from VersiCOMM command mode.


Beginning with version 1.1(109), SENDZ includes support for the YMODEM protocol and remote execution. See the /YMODEM and /REMOTE switch descriptions below.

**Operation**

The format of the SENDZ command is:

```
>SENDZ filespec/switches 
```

SENDZ must be run from the VersiCOMM command mode prompt. The 'filespec' may be one or more standard AMOS wildcard file specifications. For example,

```
>SENDZ *.TXT 
```

In some cases it may be necessary to initiate the ZMODEM receive program on the remote system prior to entering the SENDZ command at the VersiCOMM command mode prompt. For example, when using AMTEC+ you must specify that you wish to upload files and that you will be using the ZMODEM protocol.

After the SENDZ command is entered the program will respond with a display similar to the following:

```
sendz 1.1(105) 07-14-91 for AMOS
(c) 1990 Valley Programming Service, Inc.
All Rights Reserved

Rx buffer = 0 Tx frame = 0 Block size = 512
Tx window = 31744 ZCRCQ spacing = 8192

Transferring FOX.TXT 12802 bytes
12800 ZMODEM CRC-32 window = 4610
FOX.TXT transfer complete

Sendz exit
```

When using the /YMODEM option you must first start the YMODEM receive program on the remote system, prior to entering the SENDZ/YMODEM command at the VersiCOMM command mode prompt.

### Switches

The following option switches are available:

/APPEND	Append to existing destination file.
/ASCII	Change NL to CR/LF.
/BINARY	Binary file transfer override (default).
/BLOCK:n	Limit subpacket length to n bytes.
/DELETE	Delete existing file, if any.
/ESCAPE	Escape all control characters.
/FRAME:n	Limit frame length to n bytes.
/FULL	Send Full pathname.
/LONGER	Send file if source longer.
/OLDCRC	Use 16 bit CRC instead of 32 bit CRC.
/ONEK	Send 1024 byte subpackets.
/PROTECT	Protect existing destination file.
/REMOTE	Enable use at AMOS command level as a remote command
/REPLACE	Replace existing file, skip if not present at remote.
/RESUME	Resume/Recover interrupted file transfer.
/SLASH	Change '.' to '/' in pathnames.
/TIMEOUT:n	Change timeout to n tenths of a sec.
/WINDOW:n	Window is n bytes.
/YMODEM	Use YMODEM batch protocol

## Errors

### ?SENDZ must be run from VersiCOMM command mode

SENDZ cannot be run from the AMOS command level. It must be run from the VersiCOMM command mode prompt.

### ?RCLOCK must be installed

SENDZ requires that the VersiCOMM clock, RCLOCK, be installed in SYSTEM memory. Refer to the VersiCOMM INSTALLATION MANUAL for more information.

### ?Output specification not allowed

SENDZ cannot process an output file specification in its command line.

### ?Aborting ZMODEM link

SENDZ has aborted the ZMODEM link because of a user initiated abort, or an error.

### ?Could not open 'filename'

The file specified could not be opened. Usually occurs because the file is a random file or a protection violation.

### %Skipping 'filename'

The file specified will not be transferred. Usually occurs because the file exists on the remote system and the delete switch was not specified.

### ERROR in 'procedure-name'

An error occurred in the procedure specified. Usually the nature of the error will be described in a preceding message.

### Retry xx:"Bad CRC"

A transmission error has occurred. Can be the result of line noise or the sending system overrunning the receiving system.

### Retry xx: zgethdr TIMEOUT

SENDZ did not receive a frame header when expected. This error will occur if the link with the remote system is broken. It can also occur in combination with other errors when the sending system overruns the receiving system.

## Receiving Files

Following is a list of the commands that can be used to receive a file using VersiCOMM-Plus:

SAVE	Capture displayed data to disk.
RECEIVE	Receive a sequential file without protocol.
RETRV	Retrieve AMOS file(s) from another Alpha Micro using an MNP-based proprietary protocol.

RECX	Receive a sequential file using XMODEM or XMODEM 1K protocol.
RX	A version of RCX adapted for remote use.
RE CZ	Receive sequential file(s) using the ZMODEM protocol.
RE CZ/REMOTE	RE CZ with switch for remote use.
RE CZ/YMODEM	Receive sequential file(s) using the YMODEM protocol.

## SAVE

Perhaps the easiest way to save a printable file is to use the SAVE command to save the file as it is output by the remote system. If you wanted to copy a BASIC program from another system you could TYPE the file on the remote system and save it on your system as it is displayed.

## RECEIVE

This command receives sequential files "without protocol," including LIT and RUN files from other Alpha Micros. Note that if you are going to receive a file with "binary" data, i.e. data using all eight bits of the byte, it is necessary to set STRIP to OFF.

The usual procedure would be for you to run a program on the remote computer that would "download a file without protocol." If you intend to transfer binary data the remote program must output a literal copy of the file, and not do any fancy output manipulations such as expanding TAB characters. (The AMUS download program satisfies this requirement.) Such a program will normally ask you for the name of the file you want to download and then give you at least a few seconds to switch to VersiCOMM command mode and type "RECEIVE filename".

Once it is executing RECEIVE will display the message "Waiting..". It will wait up to approximately 60 seconds for the remote computer to start to transmit the file. When the first characters are received from the remote computer the message "Receiving" will be displayed. Each dot that follows represents a block of data. When the transfer is complete a message will be displayed letting you know that the file has been received and closed. If the transfer has been between Alpha Micros and the remote computer's download program outputs a literal image of the file, the DIR/H command can be used to verify that the hash total for the received file is correct.

The program MINSND.M68, included with AMOS/L VersiCOMM, is a simple send program that can be used to download files from a remote Alpha.

## RETRV

RETRV is a VersiCOMM-Plus program that will retrieve any type of AMOS file from a remote Alpha Micro with CRC16 error detection and correction. Throughput to small or heavily burdened systems can be significantly increased by using *FastQ*, our enhanced enhanced serial communications system.

Like TRANS, RETRV increases the effective speed that data is transferred by using *FasTrans* Compression or Adaptive Data Compression. Virtually all (non-compressed) files are compressible, not just files with many repeated characters. For example, data such as

this text would be compressed approximately 40%-50% by Adaptive compression and more by *FasTrans* compression.

RETRV requires the remote computer have resident a companion module, TNZA, to handle the remote end of the transfer. TNZA must be executable from the account in which you are logged on the remote computer. THE PROGRAM TNZA MUST NOT BE RENAMED!

Once a link has been established using VersiCOMM, RETRV can be run at AMOS command level or directly from VersiCOMM.

RETRV accepts wildcards and output filespecs similar in operation to the AMOS COPY command. It can transfer all types of AMOS files including text and binary sequential files, and contiguous (random) files. If needed, RETRV will create an account on the local system if the user is logged into the operator account of the local system.

RETRV uses the *FastQ* enhanced serial communications system if the file RETRV.INI exists (on the local system) in the account from which RETRV is being run, or if RETRV.INI exists in DSK0:[7,0]. The single line "FASTQ ON" should be included in this file. If the file RETRV.INI is not found, *FastQ* is not used.

The MEM: device may be specified in input or output file specifications. If a system has been booted under special circumstances, such as a warm boot, software may be transferred to user memory from the remote system if RETRV.LIT and MEM.DVR are already in local memory or can be loaded from disk.

RETRV automatically configures itself for optimal reliability when transferring files to a "slow" remote system, such as an AM1000, at 9600 bps and higher baud rates. It also adapts to the condition of the line or the burden of the receiving computer by adjusting its data rate and packet size.

In addition to the usual link level packet-oriented CRC error detection, an application level CRC hash total is calculated on the entire file and transmitted at the end of the file. If this hash total does not match that of the received file, the file is rejected. This mechanism protects against protocol and other application level software errors.

If requested, RETRV will maintain a LOG file that records the date and time of the transfer, the name of the file transferred, the total number of files transferred during the session, the total number of retries and the average transfer rate.

RETRV does not require immediate acknowledgment of transferred packets. This increases throughput on satellite links, or other links with line turnaround delays. The number of outstanding packets can be adjusted from as few as one to as many as eight packets.

RETRV does not expend the CPU time of the local system unnecessarily. CPU time is conserved unless the transfer is at 9600 bps or a higher rate, and the receiving system is an older system such as an AM1000.

## Operation

RETRV may be run from AMOS command level, or directly from VersiCOMM. The general form of the RETRV command is:

```
RETRV {local-fspec=} remote-fspec {(terminal-name)}/{switches}
```

Type RETRV followed by the name of the file you wish to transfer followed by the name of the terminal, enclosed in parentheses, associated with your modem or other RS-232 link to the other computer.

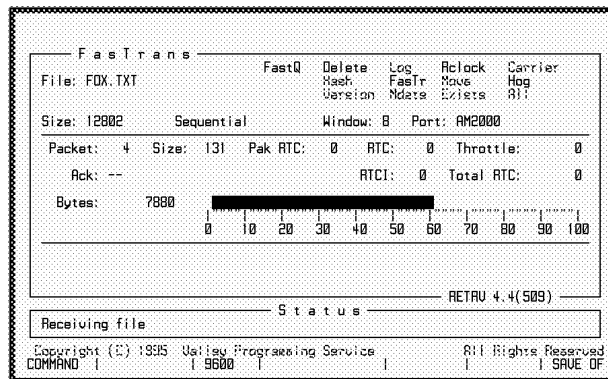
If you do not specify a terminal-name RETRV uses MODEM1, unless you are running RETRV from the VersiCOMM-Plus command prompt, in which case RETRV will use the modem that VersiCOMM is already using.

Following are examples of RETRV command lines with and without a terminal-name specified:

```
RETRV FILE.TXT
```

```
RETRV FILE.TXT(MODEM3)/D
```

When RETRV executes it responds with a display similar to the TRANS File Transfer Display.



The upper section of the window displays the context of the file being transferred. The following information is provided:

File	File specification of local file.
Size	Size of file transmitted in bytes.
Sequential -	Type of file. "Random" is an alternate type of file.
Window	Is the number of packets that can be outstanding without an ack.
Port	The terminal definition (TRMDEF) of the communications port that
#00001	A running total of the number of files transmitted.

The middle section of the window displays the current state of the transfer, including:

Packet	Sequence number, ranging from 0 to 255, of the last packet transmitted
Ack	Not displayed.
Size	Number of data bytes in the packet.
Pak RTC	Retry count for the current packet.
RTC	Retry count for the current file.
Total RTC	Cumulative retry count for all files.
RTCI	Retry count index.
Throttle	Percentage reduction in speed of transmission.
Bytes	Number of data bytes received and output to the local file.


The middle section of the window also includes a bar graph that indicates the percentage of the file that has been transferred.

The lower sections of the display lists status messages and, upon completion of the transfer, the following statistics:

- The number of files transferred.
- The total number of bytes in the files transferred.
- The percentage compression achieved for all files transferred.
- The number of bytes that did not need to be transferred as a result of data compression.
- The average transfer rate.


The average transfer rate is calculated by dividing the total bytes in the files transferred by the total time for the transfer. The actual data rate is higher because of the additional bytes that must be transmitted as part of the file transfer protocol.

RETRV supports wildcarding and other features supported by the AMOS system file, CMDLIN.SYS. Multiple files of differing types may be specified and sent with a single command. In general conventions followed by the AMOS command COPY apply to RETRV. For example, if you were logged in the OPR: account on both the local and remote system the following command,


```
RETRV *.NEW[300,1]=*.BAS[200,70] 
```

would transfer all of the BAS files in [200,70] of the remote system to [300,1] of the local system, renaming them to NEW files and creating the account [300,1] if necessary.

If you are logged into the operator account of the remote system, files can be read from several different accounts on the remote system and sent to one account on the local system. For example,

```
RETRV =*.BAS[200,50],*.BSI[200,0] 
```

would transfer the files specified to whatever account the user is logged into on the local system. While,

```
RETRV *.BAS[200,50],*.BSI[200,0] 
```

would attempt to transfer the BAS files to [200,50], and the BSI files to [200,0] of the local system. This would require that you be logged into one of the [200,x] accounts on the local system.

With the implementation of CMDLIN.SYS wildcarding the placement of switches in the RETRV command line is much more flexible than in earlier versions of RETRV. When using RETRV from AMOS command level, switches may be placed anywhere in the command line.

In addition to wildcarding, CMDLIN.SYS supports the specification of files via an "indirect file." That is, the files to be transferred by RETRV can be specified in a file. For example, if the file "FILES.IND" contained the following lines:

```
FILE1.DAT,  
FILE2.DAT,  
FILE3.DAT/D
```

The command RETRV @FILES would be equivalent to RETRV FILE1.DAT,FILE2.DAT,FILE3.DAT/D. Note that even though the file specifications are on separate lines of the file, they are appended to form one file specification.

### Switches

/FASTRANS or /NOFASTRANS

If adequate memory is available on the remote system, RETRV uses ***FasTrans*** data compression as a default. This type of compression normally reduces the number of characters transferred by 45% to 75% or more. It compresses virtually all (uncompressed) files. A text file such as this would be compressed by about 65%!

A fixed /PACKET size of 9000 is used when ***FasTrans*** compression is specified. A smaller packet may be used, but this will reduce throughput.

If insufficient memory is available to use ***FasTrans*** compression, then Adaptive compression is used. To force the use of Adaptive compression use the switch /NOFAST.

/ADAPT or /NOADAPT

RETRV will fallback to Adaptive compression when ***FasTrans*** compression cannot be used. This type of compression reduces the number of characters transferred by 25% to 50% or more. It compresses virtually all (uncompressed) files. A text file such as this would be compressed by about 40%!

If the link speed and receiving computer burden permit, a packet size of about 500 may improve throughput by another 10% or so. See the discussion of the /PACKET and /MAXIMUM switches below.



When /NOADAPT is specified, RETRV reverts to "run length" compression, which is much less CPU intensive, but efficient only on those files that contain many repeated characters, such as nulls in a partially full random file.

#### /LOG or /NOLOG

If specified, RETRV will log the name of the file transferred and the number of retries that occurred during the transfer, in the file RETRV.LOG. If the log file already exists new entries will be appended. The entire filespec, including device and ppn, is listed in the log file entry. If the /INFO switch is set (the default), the effective rate of the transfer will also be logged.

#### /RETRY:n

Controls the number of times RETRV will attempt to establish a remote link (n/2 times), resend packets (n times) and how many consecutive retransmissions of different packets may occur without a valid acknowledgment (n packets). If

/R is specified but n is not specified, 50 is used. If /R is not specified, 12 is used.

#### /DELETE or /NODELETE

The /DELETE switch indicates that a file should be retrieved from the remote system, even if it already exists on the local system. When this switch is specified, if the file does exist on the local system, it is renamed to have an extension of ".FTB" PRIOR to the actual transfer. This is important to remember if the transfer is interrupted!

#### /UPDATE or /NOUPDATE also /HASH or /NOHASH

The /UPDATE or /HASH switch indicates that a file should be retrieved from the remote system if it does not exist on the local system, or if it does not have the same hash total as the file on the local system. If the file on the local system has the same hash total, the remote file is not transferred. If the remote file is transferred to the local system, the original local file is renamed with an extension of ".FTB" PRIOR to the transfer. The /UPDATE and /HASH switches are functionally identical.

#### /VERSION or /NOVERSION

This switch indicates that a file should be retrieved from the remote system if it does not exist on the local system, or if the (remote) file has a later version number than the file on the local system. If the file on the local system has the same or later version number, the remote file is not transferred. If the remote file is transferred to the local system, the original local file is renamed with an extension of ".FTB" PRIOR to the transfer.

#### /MDATE or /NOMDATE

The /MDATE switch indicates that a file should be retrieved from the remote system if it does not exist on the local system, or if the (remote) file has a 'modified' date later than the file on the local system. NOTE: BOTH THE LOCAL AND THE REMOTE FILE MUST BE DATE STAMPED FOR THE TRANSFER TO TAKE PLACE. If the remote file is transferred to the local system, the original local file is renamed with an extension of ".FTB" PRIOR to the transfer.

**/EXISTS or /NOEXISTS**

This switch indicates that a file should be retrieved from the remote system **ONLY** if it exists on the local system. It may be used in combination with other switches such as /HASH, /VERSION, and /MDATE.

**/RANDOM or /NORANDOM**

The /RANDOM switch indicates that **ONLY** random files in the wildcard set are to be retrieved. Note that /NORANDOM means not "only random files."

**/SEQUENTIAL or /NOSEQUENTIAL**

The /SEQUENTIAL switch indicates that **ONLY** sequential files in the wildcard set are to be retrieved. Note that /NOSEQUENTIAL means not "only sequential files."

**/MOVE or /NOMOVE**

When this switch is specified, the transmitted file (the remote file) is renamed to have an extension of ".FTB" when the transfer has been completed.

**/WINDOW:n**

RETRV normally does not require immediate acknowledgment of transmitted packets. Up to 8 packets may be outstanding. In most situations this increases throughput, especially on links that have delays, such as satellite links. The

/W switch enables the user to specify the number of packets that may be outstanding, ranging from 1 to 8. Using /W:1 can help when the default of /W:8 overruns the local system. It can also be helpful if it is necessary to use a small packet size when transferring files via a DATA RACE or similar modem. (See the discussion of /PACKET below.)

**/HOG or NOHOG**

When RCLOCK is installed and running in SYSTEM memory on the receiving system, this switch forces the remote receiver into a high-speed "hog the CPU" mode. The receiver automatically uses this mode if the receiving system is a 68000 or 68010 and the modem port is set for 9600 bps or faster. This switch has no effect if RCLOCK is not installed.

**/PACKET:n and /MAXIMUM:n**

The /P:n and the /M:n switches have similar functions. The /P switch permits the user to specify a **FIXED** packet length, while the /M switch is used to specify a maximum **VARIABLE** packet length. A packet size of up to 9000 bytes may be specified. When Adaptive compression is selected, RETRV uses a variable packet size based on the number of retries that have recently occurred. When ***FastTrans*** compression is selected, a fixed packet size is used.

When using DATA RACE modems or similar modems with high and low speed channels, use of the /P switch prevents acknowledgments from choking the slow channel, due to many

small packets being transmitted on the high speed channel. Use a fixed packet size of about 400.

Use of a large packet size decreases the protocol overhead, but increases the input overhead on the receiving system.

When the receiving computer is very burdened, or the line very noisy, the /SPRINT switch will often get a file through faster than more conventional configurations. A very small packet size may be required in unusual circumstances.

#### /SPRINT

This is a convenience switch that is equivalent to:

/WINDOW:1/PACKET:100/HOG

Use this configuration when transferring a file at high speed to a slow or burdened system.

#### /CARRIER

When this switch is specified, a change in the state of DCD (usually pin 8) from the state detected when RETRV started results in RETRV aborting the transfer.

#### /INFO or /NOINFO

This switch results in the display of information about the amount of compression achieved (when using adaptive compression) and the actual throughput of the transmission, including wildcarding and protocol overhead. Note that throughput info may not be accurate on very short transfers.

#### /EXCLUSIVE or /NOEXCLUSIVE

This switch determines whether the file on the transmitting system is opened for exclusive or shared access.

All switches are global. They affect all files regardless of where they are placed in the command line.

When FasTrans compression is used the default configuration is:

```
/FASTRANS/NOFILELIST/NOLOG/RETRY:12/NODELETE/NOUPDATE/WINDOW:8
/PACKET:9000/NOSPRINT/NOMOVE/NOMAXIMUM/NOKILL/NOCARRIER/INFO
/NOVERSION/NOMDATE/NOEXISTS/NORANDOM/NOSEQUENTIAL/EXCLUSIVE.
```

When Adaptive compression is used the default configuration is:

```
/ADAPT/NOFILELIST/NOLOG/RETRY:12/NODELETE/NOUPDATE/WINDOW:8
/NOPACKET/NOSPRINT/NOMOVE/MAXIMUM:200/NOKILL/NOCARRIER/INFO
```

/NOVERSION/NOMDATE/NOEXISTS/NORANDOM/NOSEQUENTIAL/EXCLUSIVE.

If /NOADAPT is specified /MAXIMUM:512 is used.

### **Troubleshooting**

#### **Cannot work interactively on the remote Alpha Micro.**

*Problem is not related to file transfer.*

#### **Can work interactively on the remote Alpha Micro, but cannot initiate file transfer. RETRV gives the message, "?Cannot establish remote link."**

*Check that RETRV is using the same port as that being used when working interactively on the remote system.*

*Check that an executable copy of TNZA exists on the remote system. Normally the file TNZA.LIT should reside in DSK0:[1,4] or system memory.*

*Be sure that the communications port and any intermediate device such as a modem is set for 8 data bits. Normally all ports on the Alpha Micro are set to 8 data bits when the system is booted. The comm port is set to 8 data bits when COMM is first executed, but can be changed to other configurations by using the SIZE, PARITY and STOP commands.*

#### **The file transfer proceeds normally for part of the transfer and then no packets are received and no retries are recorded. Finally RETRV gives the message, "?Link timed out"**

*Check if the communication link has been interrupted.*

*Be sure that XON/XOFF flow control has been disabled and that XON/XOFF pass-through has been enabled on the modem or any other intermediate device in the communications link. Some 3rd party interfaces also support XON/XOFF flow control at the port level. Be sure this is disabled.*

#### **When the file transfer is initiated, RETRV outputs the message "%Adaptive compression not supported on remote."**

*This message results when the TNZA.LIT program on the remote system is a version that does not support Adaptive Data Compression. Often this occurs when TNZA is inadvertently replaced when the operating system is updated. TNZA.LIT is included as part of the operating system, but usually is an earlier version than that being distributed with VersiCOMM.*

### **Errors**

#### **%Adaptive compression not supported on remote**

The remote server TNZA.LIT is not current and does not support Adaptive compression.

**?Assigned FAX job not running FAX program**

The FAX job was not running FAXSER or other FAXxxx program.

**?Bad table**

A table used in *FasTrans* compression was not valid.

**?Cannot output to OPR:BADBLK.SYS**

RETRV cannot transfer a file to OPR:BADBLK.SYS.

**?Cannot RETRV an 'empty' sequential file**

The file specified is empty, i.e. it contains no data bytes.

**?Communications port is in use**

The terminal selected for use in communications (usually a modem) is already in use by a VPS communications program.

**?Could not add PPN**

In order to accomplish the transfer it would be necessary to add a new PPN on the remote system, and an error occurred when this was attempted. Usually this is the result of a "protection violation" on the remote system.

**?Could not establish remote link**

The remote computer either did not respond to repeated commands to initiate the link. Either quality of the communications line was very poor or TNZA was not executable on the remote computer.

**?Could not find PSEUDO driver**

RETRV uses the PSEUDO driver for communications during file transfers. This driver is part of the monitor and should always be found.

**?CRC sum of local file and remote file did not match**

The hash total of the file transmitted does not match that of the file received. Notify Valley Programming Service of ANY occurrence of the message.

**?Error opening file on remote system**

RETRV could not OPEN the file specified on the remote system.

**?Error opening file on this system**

An error occurred in preparation for opening the receive file on the local system.

**?Error opening RANDOM file**

An error occurred on the local system opening the receive file for random I/O.

**?Error opening SEQUENTIAL file**

An error occurred on the local system opening the receive file for sequential output.

**?Ersatz device is not allowed in output filespec**

Since an ersatz device may not have the same definition on a local and remote system, ersatz device specifications are not valid in an output filespec.

**%FasTrans could not be used on remote - using adaptive compression**

*FasTrans* compression could not be used on the remote system. This could be because there is insufficient memory for TNZA.LIT (the remote server) to use *FasTrans* compression, or TNZA.LIT is not current and does not support *FasTrans* compression.

**?FAXCOM not responding, FAXSER not restarted...**

FAXCOM did not respond when RETRV attempted to restart FAXSER.

**?FAX job not in CMD or COM state**

The FAX job was not running FAXSER, FAXCMD or FAXCOM.

**?File not resident on this system--exists switch specified**

The /EXISTS switch was specified and the file is not being retrieved because it does not exist on the local system.

**?File resident on local system--date is the same or later**

The file is not being retrieved from the remote system because the /MDATE switch was specified and the date stamp on the local file is the same or is later than the date stamp on the remote file.

**?File resident on this system--date not available**

The /MDATE switch requires that the remote file being retrieved and the local file, if it exists, have a date stamp.

**?File resident on local system--delete switch not specified**

The delete switch /D must be specified in order to retrieve a file which already exists in the account into which you are logged on the receiving computer.

**?File resident on local system--hash codes are the same**

This message is output when the /UPDATE or /HASH switch has been specified and a file is skipped because the transfer of the file would be redundant.

**?File resident on local system--version is the same or later**

This message is output when the /VERSION switch has been specified and a file is skipped because the transfer of the file would be redundant.

**?Link aborted by remote**

An error occurred on the remote system that required the transfer to be terminated.

**?Link timed out**

No data was received by RETRV for approximately 60 seconds.

**?No data received from remote**

RETRV expected to receive data packets and none were received.

**?No date stamp on remote file**

The /MDATE switch was specified, but there is no date stamp on the remote file.

**?Not enough space to allocate random file**

You tried to transfer a random file that would not fit on the local computer.

**?No files on the remote system meet your criteria**

RETRV could not locate any file matching the file specification(s) in the RETRV command line, with the switches (if any) specified. This will be the last message displayed, and may overwrite a previous message, such as "?File resident on local system--delete switch not specified."

**?OIP timeout**

The terminal's output in progress flag did not clear when expected.

**?Output error -- check if device full**

An error occurred while attempting to output data to the receive file.

**?QUERY switch has not been implemented in RETRV**

You cannot specify the /Q switch when using RETRV.

**?Receive buffer overflow**

A receive buffer overflow trap error occurred.

**?Remote transmit program TNZA is active**

If the communications link is interrupted while RETRVing a file and then reestablished, this message may be displayed if you immediately attempt to RETRV a file.

**?Retry count exceeded--line too noisy**

RETRV tried to transfer the same block more than 12 times, or the currently defined maximum retry count.

**?Terminal not found**

The terminal specified, or the default terminal MODEM1, was not defined in your system as a terminal.

**?Unknown error code from remote system**

An unknown error occurred on the remote system.

**RECX AND RX**

RECX and RX are Xmodem compatible programs used to receive sequential files from another asynchronously linked computer equipped with a Xmodem compatible software package. Both CRC16 or CHECKSUM modes of operation are supported.

Normally RECX is used by an Alpha Micro user to receive a file from a remote PC, and RX is used by a PC user to receive a file from a remote Alpha Micro.

Once a link has been established using VersiCOMM, RECX can be run at AMOS command level or directly from VersiCOMM.

## RECX Operation

The format of the RECX command is:

```
.RECX filespec (terminal-name)/switches
```

To use RECX you would first use VersiCOMM to establish communications with a remote system that supports the XMODEM protocol. Next, on the remote system, you would enter the command(s) to send a file from the remote system using the XMODEM protocol. This process is often referred to as "downloading a file."

After you have started the XMODEM send on the remote system, switch to VersiCOMM command mode and type RECX followed by the name of the file you wish to transfer. For example,

```
RECX MOVCUR.TXT
```

If RECX is being run from the AMOS command level it will assume the terminal-name for the modem port is MODEM1, unless an alternate terminal-name is specified in parentheses. For example,

```
RECX MOVCUR.TXT(TERM3)
```

The program will then respond with a display of the status of the transmission similar to the following:

```
.RECX MOVCUR.TXT/D
Waiting for initial packet from sender...^C to ABORT
Transfer initiated -- using CRC-CCITT error detection
Blocks received: 7 Retries: 0
Final response from sender:
.
```

Note that the block size is 128 bytes, not the usual 512 byte AMOS block.

## RX Operation

The format of the RX command is:

```
.RX filespec/switches
```

To use RX, first use the communication program on your PC (or other computer) to establish communications with the Alpha Micro to which you want to send a file. When you have logged onto the Alpha Micro and can enter commands, type RX followed by the name of the file you want to transfer. For example,

```
.RX MOVCUR.TXT
```



RX will then output the message,

```
Waiting for initial packet from sender
```

At this point initiate the XMODEM send process using the procedure prescribed for your PC's communication software.

### Switches

- /D                    - Delete file if already resident on system.
- /S                    - Use checksum (instead of CRC) error detection.

### Errors

#### ?File already exists

The file already exists on the local computer and the delete switch was not specified.

#### ?Terminal not found

The terminal specified, or the default terminal MODEM1, was not defined in your system as a terminal.

#### ?Communications port is in use

The terminal selected for use in communications (usually a modem) is already in use by a VPS communications program.

#### ?Could not find PSEUDO driver

RECX uses the PSEUDO driver for communications during file transfers. This driver is part of the monitor and should always be found.

#### ?More than 10 attempts to start sender -- transfer canceled

The remote send program would not start sending blocks.

#### ?Synchronization error -- transfer canceled

A block was received out of sequence.

#### ?More than 10 attempts to receive block -- transfer canceled

The same block was transmitted 10 times unsuccessfully.

### RECZ

RECZ receives sequential files using the ZMODEM or YMODEM protocols. ZMODEM is generally regarded a much more robust and efficient protocol and if the sending system supports both protocols, ZMODEM should normally be used. ZMODEM was originally developed to alleviate throughput problems experienced by packet-switched network customers using the XMODEM and KERMIT file transfer protocols. It is the most capable of the file transfer protocols offered on AMTEC+.

Unlike XMODEM, it does not pad the last block of a transferred file. Much faster than KERMIT, ZMODEM uses a streaming protocol with very low overhead and a 32-bit CRC checksum. In addition, multiple files may be downloaded in a single transfer, and should the transfer be interrupted you can dial back into the system and resume the transfer at the point at which it was interrupted.

For RECZ to properly execute, RCLOCK must be installed and RECZ must be run from VersiCOMM command mode.

Beginning with version 1.1(110), RECZ includes support for the YMODEM protocol and remote execution. See the /YMODEM and /REMOTE switch descriptions below.

RECZ also now supports the *FastQ* enhanced serial communications system. RECZ uses *FastQ* if the file RECZ.INI exists (on the local system) in the account from which RECZ is being run, or if RECZ.INI exists in DSK0:[7,0]. At this time it is not necessary to place a FASTQ ON command in this file. If the file RECZ.INI is not found, *FastQ* is not used.

RECZ will attempt to prevent overrun of the receiving system using hardware flow control. If your modem supports this feature, be sure it is configured for hardware (CTS/RTS) flow control, and that the modem cable is properly constructed. Typically this means connecting 4 to 5, and 5 to 4, in addition to the usual cable configuration.


### Operation

You should initiate the ZMODEM transmit program on the remote system prior to entering the RECZ command at the VersiCOMM command mode prompt. For example, when using AMTEC+ you must specify that you wish to download files, the file(s) you want to download, and that you will be using the ZMODEM protocol.



If auto-downloading is enabled, it will not be necessary to switch to command mode and enter the RECZ command.

The format of the RECZ command is:

```
>RECZ/switches 
```

RECZ must be run from the VersiCOMM command mode prompt. No file names should be specified in the RECZ command line.

RECZ supports the resumption of an interrupted file transfer session. If a "download resume" command is available on the remote system it is not necessary to explicitly specify the "/R" switch in the RECZ command line.

If the name of a transferred file is not valid on the Alpha Micro system, for example if it's too long, RECZ renames the file to Z00001.FIL, Z00002.FIL, Z00003.FIL, etc. If Zxxxxx.FIL already exists, the next file in the Zxxxxx sequence will be used.

### Switches

The following switches are defined:

/APPEND	Append to existing destination file
/BINARY	Binary file transfer (default)
/DELETE	Delete existing file if any
/ESCAPE	Escape control characters
FLOW	Manage input buffer using RTS flow control (default)
/PARTIAL	Allow reception of partial file so can use RESUME (def)
/PROTECT	Protect existing destination file
/RESUME	Resume interrupted file transfer
/RETRY:n	Change the maximum number of retries to n
/SEGMENT	Use segmented streaming (default on 68000 and 68010)
/TIMEOUT:n	Change timeout to n tenths of sec
/WINDOW:n	Change window to n characters

## Errors

### ?RECZ must be run from VersiCOMM command mode

RECZ cannot be run from the AMOS command level. It must be run from the VersiCOMM command mode prompt.

### ?RCLOCK must be installed

RECZ requires that the VersiCOMM clock, RCLOCK, be installed in SYSTEM memory. Refer to the VersiCOMM INSTALLATION MANUAL for more information.

### ?Output specification not allowed

RECZ cannot process an output file specification in its command line.

### ?Could not establish ZMODEM link

RECZ was not able to establish a dialog with the remote system.

### ?ZMODEM link aborted

### ?Aborting ZMODEM link

The ZMODEM link was aborted at the request of the user or as a result of an error.

### 'filename' exists, skipping to next file

The file specified will not be transferred. This occurred because the file exists on the local system and the delete switch was not specified.

### ?Error opening 'filename'

The file specified could not be opened. Usually occurs because the file is a random file or as a result of a protection violation.

**%Erasing partial file**

The /NOPARTIAL switch was specified, and the transfer of the file has been interrupted prior to completion.

**Recz exit -- TIMEOUT on ZFIN**

RECZ did not receive the final (ZFIN) frame from the ZMODEM sender. Reception of the file will normally be complete.

**'Filename1' name invalid - changed to 'filename2'**

The filename supplied by the remote system is not valid on the local system and is being changed to the filename specified.

**Bad CRC**

A transmission error has occurred. Can be the result of line noise or the sending system overrunning the receiving system.

**zgethdr TIMEOUT**

RECZ did not receive a frame header when expected. This error will occur if the link with the remote system is broken. It can also occur in combination with other errors when the sending system overruns the receiving system.

**Bad data subpacket****Data subpacket too long****Garbage count exceeded****Bad escape sequence xx**

These errors normally occur as a result of the loss of transmitted data, such as when the transmitting system overruns the receiving system. Check and make sure your modem cable is constructed to support hardware flow control and that your modem has been configured for hardware flow control.

**KERMIT**

KERMIT is a public domain file transfer program that is available from the Alpha Micro User's Society and other sources. KERMIT can be retrieved from the AMUS computer using RETRV. All AMUS members are provided with an account on the AMUS Network Computer.

KERMIT is to larger computers what the XMODEM protocol is to PC's. It provides a means for transferring files between a large number of otherwise incompatible systems.

To use KERMIT, temporarily exit from VersiCOMM without disconnecting from the remote system using the EXIT command.

## BASIC Interface

---

VersiCOMM-PLUS can execute AlphaBASIC programs at VersiCOMM command level using the 'RUN' command or from a script file using the '\$RUN' command. VersiCOMM-PLUS also can:

1. pass information from the VersiCOMM command line to a USING file.
2. exchange information between USING files and AlphaBASIC.
3. use the TREAD, TWRITE and TWAIT subroutines for communicating with the modem port from AlphaBASIC.

NOTE: When using the VersiCOMM 'RUN' command on AMOS 2.x systems, be sure to use a RUN file that has been COMPILED by the compiler found on the 2.x system.

### Required Map Statements

The following MAP is required in the AlphaBASIC program, if arguments are to be passed to the program or the communication subroutines are to be used in the program.

```
MAP1  ARGBLK
      MAP2  TCHAN,B,4
      MAP2  ARG0,S,30
      MAP2  ARG1,S,30
      MAP2  ARG2,S,30
      MAP2  FLG0,B,1
      MAP2  FLG1,B,1
      MAP2  FLG2,B,1
      MAP2  FLG3,B,1
      MAP2  ARG3,S,60
      MAP2  ARG4,S,60
      MAP2  ARG5,S,60
      MAP2  ARG6,S,60
      MAP2  ARG7,S,60
      MAP2  ARG8,S,60
      MAP2  ARG9,S,60
      MAP2  ARG10,S,60
      MAP2  ARG11,S,60
      MAP2  ARG12,S,60
      MAP2  ARG13,S,60
      MAP2  ARG14,S,60
      MAP2  ARG15,S,60
      MAP2  FLG4,B,1
      MAP2  FLG5,B,1
      MAP2  FLG6,B,1
      MAP2  FLG7,B,1
      MAP2  FLG8,B,1
      MAP2  FLG9,B,1
```

```
MAP2 FLG10,B,1
MAP2 FLG11,B,1
MAP2 FLG12,B,1
MAP2 FLG13,B,1
MAP2 FLG14,B,1
MAP2 FLG15,B,1
```

## GCMARG and PCMARG XCALLS

The GCMARG and PCMARG XCALLS are used to pass the data mapped in ARGBLK to the BASIC program and then back to VersiCOMM.

The format of the calls is:

```
XCALL GCMARG,ARGBLK,TCODE      (Gets the data from VersiCOMM)
XCALL PCMARG,ARGBLK,TCODE      (Puts the data back in VersiCOMM)
```

Currently defined completion code errors for these calls are:

- |    |  |
|----|--|
| 1  | Wrong number of parameters                           |
| 2  | Could not locate VersiCOMM COMMON area               |
| 3  | ARGBLK is not mapped                                 |
| 20 | Not enough memory was available to execute the XCALL |

GCMARG would normally be called at the logical beginning of the BASIC program, and PCMARG at the logical end.

## TREAD and TWRITE XCALLS

The TREAD and TWRITE Subroutines extend the capabilities of AlphaBASIC to include file-like input/output via one or more RS-232 ports. This provides a means for receiving data from a remote computer and sending data to the remote computer directly from an AlphaBASIC program.

- |            |  |
|------------|--|
| TREAD.SBR  | Receives data from the remote computer via the port.<br>Should be loaded into system or user memory for accurate timing and efficient operation. |
| TWRITE.SBR | Sends data to the remote computer via the port.  |

The GCMARG XCALL must have been run prior to using TREAD or TWRITE.



These calls are described as sending data to and receiving data from a "remote computer." THESE CALLS CAN BE USED TO WRITE AND READ THE RESPONSES (IF ANY) FROM VIRTUALLY ANY RS-232 PERIPHERAL THAT IS CONNECTED TO THE COMM PORT. To be precise, the calls send data to and receive data from the RS-232 port that is currently in use by VersiCOMM.

For example, if a smart modem is connected to the port, and it is in the command state, these calls can be used to send commands to the modem and then read the modem's responses. If the modem is in the "on line" state then any characters sent to the port via a TWRITE are relayed by the modem to the remote equipment (computer) and any characters received from the remote equipment (computer) are relayed to the port's input buffer and can be read using TREAD.

## XCALL TWRITE

The format of this XCALL is:

```
XCALL TWRITE,TCHAN,TDATA,TCODE
```

This call sends the data defined by the AlphaBASIC variable, expression or constant TDATA to the communications equipment (usually a modem) connected to the communications terminal/port associated with TCHAN and previously opened by a GCMARG call.

The parameters used in this XCALL are defined as follows:

TCHAN	- The "channel" (i.e. port) to which the data is to be written. This parameter is set by the GCMARG call.
TDATA	- This is the data that is sent to the communications equipment via the port selected by TCHAN. It can be any valid AlphaBASIC expression, variable or constant. For all types of data, except strings, all bytes comprising the data, as defined by AlphaBASIC, will be written to the port. If the data is a string only those characters up to the string's terminating null (or its maximum size) will be written.
TCODE	- This is a floating point variable that contains a completion code upon return from the XCALL. If TCODE=0 no errors in operation were detected.

Currently defined completion code errors for the TWRITE call are:

1	Wrong number of parameters in TWRITE call
2	TCHAN has not been set by a GCMARG call

### Examples of TWRITE Calls

The following calls are equivalent:

```
STRING'VAR="A command"
XCALL TWRITE,TCHAN,STRING'VAR,TCODE
```

and,

```
XCALL TWRITE,TCHAN,"A command",TCODE
```

If data was mapped as MAP1 EXAMPLE,X,10,"ABCDEF"

```
XCALL TWRITE,TCHAN,EXAMPLE,TCODE
```

would send "ABCDEF" and then four nulls to the remote computer. However if the data was mapped as MAP1 EXAMPLE,S,10,"ABCDEF", the same TWRITE call would only send "ABCDEF".

## XCALL TREAD

There are three variations of this call:

```
XCALL TREAD, TCHAN, TDATA, TCODE
```

```
XCALL TREAD, TCHAN, TDATA, TCODE, TSTART, TEND
```

```
XCALL TREAD, TCHAN, TDATA, TCODE, TSTART, TEND, TFIRST, TLAST
```

This call reads data received by the communications equipment connected to the port selected by TCHAN (and stored in the input buffer of that terminal/port) and transfers the data to the variable TDATA. TREAD reads and transfers data until either the maximum size of the variable specified is reached or no additional characters are received for a specified interval of time, or a termination character is received. Unread characters remain in the input buffer of the selected terminal/port until the next TREAD.

If you do not wish to use data left in the buffer, in most cases you should still TREAD it to clear the buffer. This procedure simplifies processing the next valid response from the remote computer and also keeps the input buffer from overflowing.

TREAD versions 1.0(110) and later include the following enhancements:

1. Hardware based input flow control is enabled. (See note below.)
2. The RTS output (usually pin 5) will be dropped on a normal exit from TREAD. When called, TREAD automatically raises RTS if no characters are received. If another SBR is used to read data following a TREAD, be sure to use XCALL SETPIN to raise RTS.
3. RCLOCK will be used if it is available.
4. Maximum CPU utilization will be enabled on systems using 68000 or 68010 CPUs.

The TCODE parameter may be used to alter the operations of TREAD. If TCODE=1 on entry, then flow control will be disabled. If TCODE=4 then DTR will be dropped on exit from the subroutines. TCODE=2 is reserved.

For flow control to be fully effective you must:

1. Configure the modem/device with which you are communicating to respond to hardware based flow control signals.
2. Construct your modem/device cable to support hardware based flow control.
3. Use a serial interface that supports automatic RTS toggling.

There are many other factors that can effect flow control performance. If you need additional information, see the article concerning flow control in the July 1992 issue of the *Alpha Micro Technical Journal*.

The parameters used in this XCALL are defined as follows:

TCHAN	The "channel" (i.e. port) from which the data is to be read. This parameter is set by the GCMARG call.
-------	---



TDATA	This is the AlphaBASIC variable into which received data is transferred.
TCODE	This is a floating point variable that contains a completion code upon return from the XCALL. If TCODE=0 no errors in operation were detected.
TSTART	This is a floating point variable equal to the number of seconds TREAD will wait to <i>start</i> reception of an expected response from the remote computer via the selected communications channel. If TREAD receives data, then the TSTART timeout is no longer valid, and the TEND timeout becomes effective. If TSTART is included in the argument list, TEND should also be included. The floating point argument is significant to 0.05 of a second. An argument less than 0.05 would be interpreted as a zero. A zero DOES NOT result in a default value being used and is valid as an argument. If neither TSTART, or TEND is included in the argument list (or TFIRST or TLAST), TSTART will default to about 15 seconds. The value specified for TSTART should be considered approximate.
TEND	This is a floating point variable equal to the minimum number of seconds TREAD will wait before assuming that all characters in a given response have been received. If TEND is included in the argument list, TSTART should also be included. The floating point argument is significant to 0.05 of a second. An argument less than 0.05 would be interpreted as a zero. A zero DOES NOT result in a default value being used and is valid as an argument. If neither TSTART, or TEND is included in the argument list (or TFIRST or TLAST), TEND will default to about 1 second. The value specified for TEND should be considered approximate.
TFIRST	A single character string that specifies the beginning of the data stream you wish to read. All data read from the port will be discarded until the specified "character" is received. If a null ("") is specified data will be read beginning with the first character received. If no character is received during the TSTART time, the XCALL will timeout. If TFIRST is included in the argument list, TLAST should be included in the argument list.
TLAST	TLAST is a single character string that terminates the end of the data stream you wish to read. Data from the port will be read until the specified "character" is received. If a null ("") is specified data will be read until there is a time lapse greater than the TEND time interval specified. If TFIRST is included in the argument list, TLAST should be included in the argument list.

Currently defined completion code errors for the TREAD call are:

- 1 Wrong number of parameters in TREAD call
- 2 TCHAN has not been set by GCMARG call
- 3 TSTART was not floating point
- 4 TEND was not floating point
- 5 TREAD timed out waiting for remote response
- 6 TFIRST not a string
- 7 TFIRST size not one
- 8 TLAST not a string
- 9 TLAST size not one
- 20 Not enough memory was available to execute the XCALL



The TEND parameter can also be used as a timing device to guarantee that a certain amount of time will pass after a certain response is received, before another command or message is sent to the communications equipment.

Using 0.0 for both TSTART and TEND results in the port buffer being "scanned" for whatever is there without waiting. Any data in the buffer will be read. If no data is present, TREAD will immediately exit without an error being indicated.

### Examples of the TREAD Call

The following calls are equivalent:

```
TSTART=15.0:TEND=1.0
XCALL TREAD,TCHAN,RESPONSE,TCODE,TSTART,TEND
```

and,

```
XCALL TREAD,TCHAN,RESPONSE,TCODE
```

The following calls are equivalent:

```
TSTART=10:TEND=3
XCALL TREAD,TCHAN,RESPONSE,TCODE,TSTART,TEND
```

and,

```
XCALL TREAD,TCHAN,RESPONSE,TCODE,10.0,3.0
```

The following TREAD would capture the data between an STX and ETX:

```
XCALL TREAD,TCHAN,RESPONSE,TCODE,10.0,3.0,CHR$(2),CHR$(3)
```

The STX and ETX characters would be included in the response and could be edited using "RESPONSE=RESPONSE[2,-2]".

### XCALL TWAIT

The TWAIT subroutine is used to wait for a specified group of ASCII characters to be received at the communications port. TWAIT reads the response in the same manner as TREAD, but does not save the data received. The format of the call is:

`XCALL TWAIT, TCHAN, WSTRING, WTIME, WFLAG, TCODE`

TWAIT versions 1.0(102) and later include the following enhancements:

1. Hardware based input flow control is enabled.
2. The RTS output (usually pin 5) will be dropped on a normal exit from TWAIT. When called, TWAIT automatically raises RTS if no characters are received. If another SBR is used to read data following a TWAIT, be sure to use XCALL SETPIN to raise RTS.
3. RCLOCK will be used if it is available.
4. Maximum CPU utilization will be enabled on systems using 68000 or 68010 CPUs.

The TCODE parameter may be used to alter the operations of TWAIT. If TCODE=1 on entry, then flow control will be disabled. If TCODE=4 then DTR will be dropped on exit from the subroutines. TCODE=2 is reserved.

The parameters used in this XCALL are defined as follows:

TCHAN	The "channel" (i.e. port) to which the data is to be written. This parameter is set by the GCMARG call.
WSTRING	Is the string that is expected to be received.
WTIME	Is the time to wait for the string in floating point seconds.
WFLAG	An operational flag. If WFLAG=1 WSTRING is displayed. If WFLAG=2 characters that are received while waiting are displayed. WFLAG=3 sets WFLAG=1 and WFLAG=2.
TCODE	This is a floating point variable that contains a completion code upon return from the XCALL. If TCODE=0 no errors in operation were detected.

Completion code errors for the TWAIT call are:

1	Wait timed out
2	TCHAN has not been set by GCMARG call
3	WSTRING not unformatted or string
4	WTIME was not floating point
5	Incorrect number of arguments
6	WFLAG was not floating point
10	Wait aborted by Control-C
20	Not enough memory was available to execute the XCALL

## Examples of the TWAIT Call

The following XCALL waits 30 seconds for the string "CONNECT":

```
WSTRING="CONNECT"
WTIME=30.0
WFLAG=3
XCALL TWAIT,TCHAN,WSTRING,WTIME,WFLAG,TCODE
IF TCODE<>0 PRINT"TWAIT ERROR ";TCODE
```

The string for which we are waiting and characters received while waiting are displayed:

```
*WAITING FOR "CONNECT"           (Output by TWAIT)

CONNECT                           (Received at comm port)
*WAIT CLEARED*                   (Output by TWAIT)
```

The following would be displayed if "CONNECT" was not received:

```
*WAITING FOR "CONNECT"           (Output by TWAIT)

*WAIT TIMED OUT*                 (Output by TWAIT)
TWAIT ERROR 1                    (Output by BASIC application)
```

The TWAIT XCALL is functionally equivalent to the \$WAIT command used in VersiCOMM script files.

## GETPIN XCALL

The GETPIN subroutine is used to obtain the status of the available RS-232 inputs for the port specified. The format of the call is:

```
XCALL GETPIN,TRMDEF,STATUS,MASK,CODE
```

The parameters used in this XCALL are defined as follows:

TRMDEF	A six character string that specifies the modem port for which status data will be obtained. If TRMDEF is null, GETPIN will check if it is being run from an AlphaBASIC program that is being run from VersiCOMM. If so, GETPIN will use the modem TRMDEF that is being used by VersiCOMM.
STATUS	A four byte binary variable that is used to return the state of each available input. Each bit of STATUS represents a signal in the RS-232 standard.
MASK	- four byte binary variable that returns a "mask" of inputs and out outputs that are available on the port specified. Each bit of MASK represents a signal in the RS-232 standard.
CODE	This is a floating point variable that contains a completion code upon return from the XCALL. If CODE=0 no errors in operation were detected.

Completion code errors for the GETPIN call are:

1	Incorrect number of arguments
2	TRMDEF is not defined as a string
3	TRMDEF argument is too large
4	TRMDEF could not be found
5	STATUS size is less than four bytes
6	MASK size is less than four bytes
20	Not enough memory was available to execute the XCALL

The program GETPIN.BAS, included on the VersiCOMM release tape in account [7,121], illustrates the use of GETPIN.SBR.

## SETPIN XCALL

The SETPIN subroutine is used to set the status of the available RS-232 outputs for the port specified. The format of the call is:

```
XCALL SETPIN,TRMDEF,STATUS,CODE
```

The parameters used in this XCALL are defined as follows:

TRMDEF	A six character string that specifies the modem port for which status data will be obtained. If TRMDEF is null, SETPIN will check if it is being run from an AlphaBASIC program that is being run from VersiCOMM. If so, SETPIN will use the modem TRMDEF that is being used by VersiCOMM.
STATUS	A four byte binary variable that is used to set the state of each available output. Each bit of STATUS represents a signal in the RS-232 standard.
CODE	This is a floating point variable that contains a completion code upon return from the XCALL. If CODE=0 no errors in operation were detected.

Completion code errors for the SETPIN call are:

1	Incorrect number of arguments
2	TRMDEF is not defined as a string
3	TRMDEF argument is too large
4	TRMDEF could not be found
5	STATUS size is less than four bytes
20	Not enough memory was available to execute the XCALL

The program SETPIN.BAS, included on the VersiCOMM release tape in account [7,121], illustrates the use of SETPIN.SBR.

## ARGn and FLGn Values

VersiCOMM maintains a set of three 30 character strings and thirteen 60 character strings, and a set of sixteen single byte binary flags.

The strings are referred to as ARG0, ARG1, etc., up to ARG15. Similarly the binary flags are referred to as FLG0 to FLG15.

The value of ARG0 - ARG15 can be set from the VersiCOMM command line, script file, or from within AlphaBASIC running under the control of a VersiCOMM script file. The value of FLG0 - FLG15 can be set from a script file, or from within AlphaBASIC running under the control of a VersiCOMM script file.

To set the ARGn values from the VersiCOMM command line use the following format:

```
COMM USING talk-file WITH arg0 arg1 arg2 ... arg15
```

For example,

```
COMM USING SENDIT WITH FILE1.LST
```

To set the ARGn or FLGn values within a VersiCOMM script file use the SETARG and SETFLG commands:

```
$SETARG 0 FILE1.LST
$SETFLG 0 128
```

The INIARG.TLK file is a special script file that can be used to set the values of ARGn and FLGn and perform other application oriented initializations. ***THE INIARG.TLK FILE IS ERASED BY VERSICOMM FOLLOWING ITS EXECUTION.***

To set the ARGn or FLGn values within AlphaBASIC assign the appropriate variable within the ARGBLK MAP to the desired value, for example:

```
ARG0 = "FILE1.LST"
FLG0 = 128
```

The values of the ARGn strings and FLGn bytes are passed from a VersiCOMM script file to an AlphaBASIC program and vice-versa using the GCMARG and PCMARG XCALLs, described earlier in this chapter.

Within a script file the value of one or more of these string arguments may replace ANY PART of a script file command by substituting its "token." The token for ARG0 is @0, ARG1 is @1, etc. Tokens greater than @9 are referred to as @A, @B, @C, @D, @E and @F:

Token	Argument
@0 - @9	ARG0 - ARG9
@A - @F	ARG10 - ARG15

If, for example, the value of ARG0 is "FILE1.LST", then the script file command "\$TRANS @0" would be equivalent to "\$TRANS FILE1.LST".



ARG3, with the token "@3", is reserved for use by the VersiCOMM DIAL command.

Following is a list of script file commands that can be used to work with ARGn and FLGn values.

INPARG	ADDFLG	INCFLG	SUBFLG	TYPEFLG
SETARG	ANDFLG	INPFLG	TRMFLG	ORFLG
TSTARG	DECFLG	SETFLG	TSTFLG	XORFLG

For more information on these commands see the VersiCOMM COMMANDS chapter of this manual.





## Terminal Emulation

---

### Virtual Terminal Emulation

Unless otherwise specified, VersiCOMM does not modify the control sequences of the terminal from which it is running, or the control sequences of the remote host. In effect, VersiCOMM "disappears" and it is as if the terminal running VersiCOMM is connected directly to the host. This is referred to as a "virtual" terminal emulation. It is the default VersiCOMM terminal emulation.

For example, if your terminal is an AM62A and you are connected to a remote Alpha Micro using VersiCOMM, other than a possible difference in baud rates the remote system would NOT need to be configured any differently than if the terminal were locally connected. The terminal definition on the remote system would still specify an AM62A terminal driver.



If your terminal generates 8-bit character codes and you plan to use a remote application that supports 8-bit character codes, you must turn the VersiCOMM STRIP switch OFF. As of VersiCOMM release 3.1 \$STRIP OFF is the default state for this switch.

When in this standard mode, VersiCOMM displays "TTY" under "Emulate" in the Command mode status display.

So what happens if you are connected to a system or device (such as protocol converter) that does not recognize your terminal's control sequences?


There are several options. In addition to their native emulation, many terminals support additional terminal emulations. In the case of the AM62A, TVI925 and ADM5 emulations are provided. If, for example, the TVI925 terminal driver was loaded in system memory (or used in a TRMDEF) you could switch to that terminal driver on the fly using CHGTDV, set your terminal to emulate a TVI925, and then dial into a system that supported the Televideo 925 terminal but not the AM62A.

Most online services offer a "TTY compatible" mode of operation that avoids the use of terminal specific control sequences.


Some services and devices require that you use a terminal or terminal emulation other than "TTY". Usually such a service will support access by a Digital Equipment VT100 terminal or emulation.

## Specifying A Terminal Emulation

The EMULATE command is used to specify a special terminal emulation. The general form of the command is:

```
EMULATE emulation-name 
```

Where "emulation-name" is a three letter code associated with a specific emulation. For example,

```
EMULATE VT1 
```

loads the emulation module and keyboard translation module for the VT100 emulation.

Emulation modules are assumed to be resident in DSK0:[7,0]. In the case of the VT100 emulation, the files DSK0:VT1.EMU[7,0] and DSK0:trmdef.VT1[7,0] are used, where "trmdef.VT1" is a keyboard translation module for the terminal running VersiCOMM. DSK0:AM62A.VT1[7,0] is an example of such a module.

The operation of VersiCOMM can be returned to the standard "TTY" emulation by specifying EMULATE OFF.

## VT100 Emulation

When "VT1" is specified in an EMULATE command, VersiCOMM-Plus emulates a Digital Equipment Corporation VT100 terminal configured as follows:

Line Feed/New Line	New Line On or Off
Autowrap	Autowrap On or Off
ANSI/VT52	ANSI Only
Cursor Keys	ANSI or Application Functions
Column	80 Column Only
Keypad	Application or Numeric
Origin	Set or Reset
Scrolling	Per Terminal Configuration
Screen	Per Terminal Configuration

The emulation supports the reduced intensity attribute and if the terminal running the emulation supports "non-space" attributes, the emulation also supports reverse, underline and blinking attributes.

The following ANSI mode control sequences are supported:

- Cursor Backward, Down, Forward, Up, Position, Position Report

- Device Attributes and Status Report
- Save and Restore Cursor
- Report and Request Terminal Parameters
- Set Top and Bottom Margins
- Erase In Line and In Display
- Horizontal Tabulation Set and Clear
- Horizontal and Vertical Position
- Index, Next Line and Reverse Index
- Set and Reset Mode
- Select Character Set - Special Graphics
- Select Graphic Rendition

Other ANSI mode control sequences are recognized but ignored.

## Lexis Emulation

The LEX terminal emulation is a slightly modified version of the VT100 terminal emulation that can be used when accessing the LEXIS<sup>2</sup> information service. The LEXIS service is designed to interface with a terminal in Block mode. Using the VersiCOMM LEXIS emulation, screen displays and system operation are very similar to that of terminal supporting Block mode.


Since many terminals in the field do not support "non-space" attributes the LEXIS emulation maps reverse video attributes into reduced intensity attributes. This maintains the LEXIS screen formatting yet "highlighted" words and areas are preserved and are easily distinguishable.

Function key generated commands can be easily implemented using the keyboard translation file associated with the LEXIS emulation and the program VPSTRN, described below. Multiple character commands can be defined as desired. These become effective when the LEXIS emulation is activated. It is not necessary to load a PFK file. The source for the AM62A LEXIS translation file, 62ALEX.M68, is included on the VersiCOMM release tape.

VersiCOMM supports the generation of a "break" from the keyboard. This enables the operator to interrupt a LEXIS search and is functionally equivalent to the STOP key.

Using VersiCOMM displayed text can be saved in disk file. In most cases it is no longer necessary to FILTER the captured text before it is printed or edited, *if* WRAP was ON when the file was captured.

If the file has lines longer than 80 characters it may still need to be processed with the FILTER program. The following command line should be used:

```
FILTER filename/W:80 
```


Where "filename" is the name of the file used to capture the text from LEXIS. The "W:80" switch wraps lines longer than 80 characters.

## Generic Terminal Emulation

A generic emulation, GEN.EMU, may be used in combination with a remotely installed generic terminal driver, GEN.TDV, to provide terminal independent operation on an arbitrary remote system.

For example, suppose VersiCOMM is used to call a remote system. If GEN.TDV is the terminal driver for the modem terminal on the remote job, then it doesn't matter what kind of terminal VersiCOMM is being run from if the Generic terminal emulation is being used. Screens will display properly, and terminal functions such as dim, reverse, etc. will work as expected, as long as the terminal being used supports the required function.

The Generic terminal emulation is specified using the command:

```
EMULATE GEN 
```

The Generic terminal emulation may be used from within MULTI.

## Additional Emulations

VersiCOMM also included emulations for the VT52 terminal and the VT220 terminal in 7-bit mode.

## Saving Displayed Text to Disk

In most cases, it is no longer necessary to FILTER text that has been captured when using a VT100 or other non-TTY emulations *if WRAP was ON* when the text was captured.

Note that captured text may not have the same formatting as it did on the screen. This is because the data is captured to the disk sequentially, but because of screen cursor addressing may not have been output to the screen so that one character physically follows the next. For example, a line near the bottom of the screen could be output *before* a line at the top of the screen.

Using the new record and playback feature of VersiCOMM-Plus it is possible to record a session and then play it back later and have it appear virtually as it was originally viewed.

## Recording Displayed Text and Playing it Back

When VersiCOMM is used to access a remote computer using a non-TTY emulation, such as the VT100 emulation, the session can be recorded and played back in simulated real-time using a file player written for the emulation. Players for the following emulations are included in the current release:


<u>Emulation</u>	<u>Name</u>	<u>Player</u>
VT100	VT1	SHWVT1
VT52	V52	SHWV52
LEXIS	LEX	SHWLEX
GENERIC	GEN	SHWGEN



Since hosts often initialize terminals during the log-on process, it may be necessary to RECORD the log-on sequence in order to capture the terminal initialization sequence, otherwise the player may not be properly initialized.

## Player Controls

The following keys can be used to control the players:

<SPACEBAR>	Pause or restart.
<ESC>	Exit.
 or 	Skip null space.

## Player Switches

The following switches are defined for all the players:










/C	Play continuously.
/I:n	Use a time interval of n. A value of n =100 is "normal."

The following switched are defined for SHWVT1, SHWV52 and SHWLEX:


/N	Turn emulation NEWLINE flag ON
/W	Turn emulation WRAP flag ON.

## Recording and Playing Back an Internet Session

Let's assume you're accessing the Internet via Netcom and you want to record your visit to a particular World Wide Web site using lynx. After logging on to Netcom:

1. Switch to VersiCOMM Command mode by pressing  and enter the command "RECORD" followed by a .
2. Enter the command "SAVE LYNXDM.SHW" followed by a . VersiCOMM should automatically switch to Conversational mode.
3. In Conversational mode start up lynx by typing "lynx" and a .
4. Access the WWW pages you want to record.
5. Exit from lynx by pressing .
6. Switch to VersiCOMM Command mode by pressing  (home) and enter the command "SAVE OFF" followed by a . (You can also turn SAVE OFF in Conversational mode by keying .)
7. Exit from VersiCOMM.

To play back the session enter the following command at AMOS command level:








```
SHWVT1 LYNXDM.SHW 
```

## Recording and Playing Back an Alpha Micro Session

To record a session on a remote Alpha Micro, the remote system must have available the GEN.TDV and GENBOX.TDV terminal drivers. If you want to record a file that will be played back on a terminal that supports box scrolling use GENBOX.TDV as the remote terminal driver, otherwise used GEN.TDV.

The SHWGEN player supports the convention that files with a "DMO" extension are for terminals that support box functions and files with a "DMX" extension are for terminals that don't.

After logging on to the remote Alpha Micro:

1. Check that you are in VersiCOMM Conversational mode, talking to the remote system. You can do a SYSTAT just to be sure. If available on the remote system, use CHGTDV or NEWTRM to check that you are using the GEN or GENBOX terminal driver.
2. Switch to VersiCOMM Command mode by pressing  and enter the command "RECORD" followed by a .
3. Enter the command "SAVE DEMO.DMO" or "SAVE DEMO.DMX" (as appropriate) followed by a . VersiCOMM should automatically switch to Conversational mode.
4. In Conversational mode start the (remote) application that you want to record.
5. When you are done recording the application, switch to VersiCOMM Command mode by pressing  (home) and enter the command "SAVE OFF" followed by a . (You can also turn SAVE OFF in Conversational mode by keying .)
6. Exit from VersiCOMM.

To play back the session enter the following command at AMOS command level:

```
SHWGEN DEMO 
```

## Keyboard Translation Files

The source code for the following terminal translation files is included in the VersiCOMM-Plus release in account [7,0]. This source code is provided for the convenience of licensed VersiCOMM users only and is not being released into the public domain.

MISVT1 M68	Miscellaneous terminal keyboard translation for VT100 emulation.
TVIVT1 M68	Teletype terminal keyboard translation for VT100 emulation.
62ALEX M68	Alpha terminal keyboard translation for LEXIS emulation.
62AVT1 M68	Alpha terminal keyboard translation for VT100 emulation.
62AVT2 M68	Alpha terminal keyboard translation for VT220 emulation.
62AV52 M68	Alpha terminal keyboard translation for VT52 emulation.

Refer to these files for information on what is transmitted to the host when a particular key is pressed. These files may be modified to change the mapping as desired. Also see the description of the VPSTRN command in the following section.

## The VPSTRN Command


The VPSTRN command is an AMOS level command that is used to modify or create two different types of keyboard translation files, standard and emulation. Standard translation files are files such as AM62A.VUX, the VUE keyboard translation file for the AM62A. Emulation translation files are files such as AM62A.VT1, the translation file for the VT100 emulation running on an AM62A. Standard translation files can also be edited by the AMOS command FIXTRN.

The format of the VPSTRN command is "VPSTRN filename" where "filename" is the name of an existing translation file, or a translation file you wish to create. For example,

```
VPSTRN TRMDEF.VT1
```

If the file does not exist VPSTRN asks if you want to create it, then asks:

```
E)mulation or S)tandard keyboard translation file? E/S:
```

Enter an "E" or an "S" followed by a , according to the type of file you wish to edit. The file formats differ slightly depending on the type. In either case VPSTRN will display:

```
Enter character to terminate sequence:
```

This single character is used to signal the end of the sequence of characters used in a translation. It **MUST** be a character that will not be used within any of the translation sequences, such as <RUB>, the "rubout" key. You cannot use a key, such as a function key, that generates more than one character. Let's say <RUB> was entered. VPSTRN then asks:

```
Enter character to indicate decimal input:
```

This single character is used within a translation sequence to indicate that you wish to specify the decimal code for a character rather than keying the character on the keyboard. The same restrictions apply to this character as apply to the termination character. In this example we'll use a plus sign, "+". Next you are asked to press the key you wish to have translated:


```
Enter the key to translate:
```

Press the keyboard character you wish to translate. If there is an existing translation it will be displayed. For example, lets say we pressed F2 and the following was displayed:

```
Translation is: <ESC>OQ
Change translation to:
```


At this point enter the sequence of keystrokes that F2 should be translated to, ending the sequence with the terminating character, <RUB> in this example. All keystrokes will be recorded, so be careful. If no translation is desired, enter the sequence terminating character as the first character in the sequence. If you need to specify a character in the sequence by its decimal value, press the key that indicates decimal input. The following prompt will be displayed:

```
Decimal value:
```

Enter the code for the character and press .

If you are working with an emulation translation file, if you specify no translation, VPSTRN will ask the following:

```
P)ass-through as is (default), or I)gnore this key? P/I:
```

A "P" or an "I" should be entered, followed by a . If "P" is selected the keycode is passed through without translation. If "I" is selected the keycode is filtered, i.e. if that key is pressed no keycode is generated.









When a translation sequence has been completed, VPSTRN asks for another key to translate. VPSTRN exits when the sequence terminating character is entered in response to the request for what key is to be translated.



## Commands

---

After using the "change mode" character (usually the <HOME> key) to switch to command mode, VersiCOMM commands may be entered at the command prompt. KEYING THE "CHANGE MODE" CHARACTER AT ANY TIME DURING COMMAND ENTRY WILL RETURN THE USER TO CONVERSATIONAL MODE. Incorrectly typed commands may be edited by using the following keys:

<DEL> or <RUB>	Delete the preceding character.
	Delete the current character.
	Erase from current character to the end of the line.
	Move to the beginning of the line.
	Move to the end of the line.
	Insert space at current cursor position.
	Move cursor left one character.
	Move cursor right one character.
	Recall previous command.

In the descriptions that follow, the general format of the command is given along with a specific example.

**>ADDFLG n m**

**>ADDFLG 0 5**

The ADDFLG command is used to add a value m to the value contained in the VersiCOMM flag n. The value to be added, m, may range in value from 0 to 255. The flag number n may be 0 to 15.

Both n and m may also be VersiCOMM flags. For example if flag #1 contains 0 and flag #2 contains 5, then ADDFLG FLG1 FLG2 is the same as ADDFLG 0 5.

These flags are included in the information that can be exchanged with AlphaBASIC using the VersiCOMM-PLUS interface.

**>AMOS amos-command****>AMOS TYPE FILE.LST**

Enables the user to execute many AMOS commands (TYPE and DIR for instance) from VersiCOMM without having to exit to the monitor. Note that this command has nothing to do with commands sent to and executed on the remote computer, but only commands that, for convenience, you want to execute locally without leaving VersiCOMM. Most AMOS commands may be executed using the AMOS command.

Because the LOG command changes several very basic system parameters, it is generally recommended that the AMOS command NOT be used to log to a new account while running VersiCOMM. If you need to LOG to another account, the only safe way is to use the EXIT command to completely exit from VersiCOMM, then LOG to the new account, and then run VersiCOMM again.

Full file specifications may be used with the TRANS and SEND command, so it is not necessary to LOG to an account to send a file (or set of files) from it.

If you inadvertently use the AMOS command to log to an account that has a START.CMD file defined, the LOG program (among other things) turns the "trace" flag off and you cannot see what you're typing. In addition the START.CMD file is not processed. To extricate yourself from the resulting situation type the sequence `␣<Q>␣`. VersiCOMM will exit and process the START.CMD file.

**>ANDFLG n m****>ANDFLG 0 5**

The ANDFLG command is used to logical "and" a value m to the value contained in the VersiCOMM flag n. The value, m, may range in value from 0 to 255. The flag number n may be 0 to 15.

Both n and m may also be VersiCOMM flags. For example if flag #1 contains 0 and flag #2 contains 5, then ANDFLG FLG1 FLG2 is the same as ANDFLG 0 5.

These flags are included in the information that can be exchanged with AlphaBASIC using the VersiCOMM-PLUS interface.

**>ANSWERBACK string****>ANSWERBACK YOUR CO UR**

When set to a non-null string the ANSWERBACK command will cause VersiCOMM to automatically respond to any Control-E received from a host and send the host the stored answerback.

**>ATTENTION**

This command invokes the "attention" modem control module "trmdef.AT0," where "trmdef" is the AMOS TRMDEF for the modem in use by VersiCOMM.

".AT0" is the default extension for this module. The MVER command may be used to specify any ASCII character "x" with a resulting extension of ".ATx".

This module is a script file that checks if the modem is responsive and sets the modem to operate properly while processing script files.

The "trmdef.AT0" is one of several modem control files that can be generated by selecting "Setup" from the VersiCOMM Dialer screen, and then "Build" from the Setup screen.

If STATUS is ON, ATTENTION displays user information on the bottom status line of the terminal, if available.

#### **>AUTOLF {ON or OFF}**

#### **>AUTOLF ON**

Enables or disables the sending of a linefeed following any carriage-return entered at the keyboard, or following an auto-answerback response. Default is OFF, that is a linefeed is NOT sent.

#### **>BACKTO label-number**

Processing of the USING file will branch backwards to the first occurrence of the label line with the label-number specified. Would normally only be used in a USING file.

#### **>BAUD rate**


#### **>BAUD 1200 or >BAUD filespec**

Changes the baud rate of the local port used for remote communication. A filespec which contains the rate may be specified instead of the rate. When this format is used the rate must be listed on the first line of the file.

#### **>BAUTO {ON or OFF}**

The BAUTO command is used to turn automatic B Plus transfers ON or OFF. The default is OFF. BAUTO should only be ON when accessing Compuserve, otherwise inadvertant B Plus protocol startups are likely to occur.

If BAUTO is ON, VersiCOMM will begin to transfer the file automatically after you have responded to the "File name for your computer:" prompt on Compuserve.

If BAUTO is OFF (the default) you will need to switch to VersiCOMM command mode and enter the command BPXFER followed by a , after you have responded to the "File name for your computer:" prompt.

Filtering of the character ASCII 05, <CONTROL><E>, MUST be turned off for automatic downloading to work. Check the TLK files used to access Compuserve and comment out any line that looks like:

```
$REMOTE 5,0      ;^E
```

If the TLK file uses the FILTER.TLK file, you will need to make a copy of the file using a different name, such as FILTR2.TLK, and after modifying the new file use that file for filtering instead.

If you use The DIALER, be sure that the Filter option is not marked in the profile for Compuserve.

**>BPXFER**

The BPXFER command is used to initiate Compuserve B Plus transfers when automatic initiation via the BAUTO switch is turned off. (See the description of BAUTO above.)

If you attempt to download a file that already exists on your system, BPXFER will ask whether you wish to Overwrite, Resume, reName, or Abort the transfer.

If you select Resume, BPXFER will attempt to resume the transfer using that portion of the file that exists on your system. A hash total is calculated for the incomplete file and this is compared to the same portion of the file on the host. If they match then the file transfer is resumed.



IF A FILE TRANSFER IS INTERRUPTED FOR ANY REASON, WHATEVER PORTION OF THE FILE THAT HAS BEEN TRANSFERRED WILL REMAIN ON YOUR SYSTEM. This is necessary to support the resume feature.

If for some reason you wish to abort a B Plus transfer, try the <ESC> key first. This attempts to notify the host of the abort request. If you use <CTRL><C>, BPXFER simply quits, without notifying the host.

If you abort a transfer and find that the host is not responding, or is still trying to send data, enter four consecutive <CTRL><P>'s, followed by a <CTRL><P> and two consecutive plus signs "++". Repeat these sequences until the host aborts.

**>BREAK {break-time}****>BREAK or >BREAK 500**

Sends a 'BREAK' to the host computer. The duration of the break will be the currently defined break-time. Initially this is 250 milliseconds. This default can be changed by specifying an argument in the command. The argument is the new break-time in milliseconds.

The break-time argument may be specified in a VersiCOMM flag. When used as an argument in a command a VersiCOMM flag is written as "FLGn", where n is the flag number, e.g. FLG0, FLG1, ..., FLG15.

This command requires the use of an Alpha Micro extended function interface driver. Extended function interface drivers are included with AMOSL 1.3D and AMOS32 1.0D and later versions of the AMOS operating system.

**>BREAKCHR ascii-code or character****>BREAKCHR 3 or >BREAKCHR ^C**

Redefines the character used for generating a 'BREAK' while in conversational mode. Initially this character is a Control-@ (null).

You may specify either the ASCII code for the character, the character itself, or a VersiCOMM flag that contains the ASCII code for the character. Control characters are indicated by a preceding '^'. VersiCOMM flags are indicated by the character string "FLGn," where n can be 0 to 15.

**>CABORT {ON or OFF}****>CABORT ON**

If CABORT is ON, VersiCOMM will exit when DCD (usually pin 8) goes into a low state.

**>CCENDTLK {ON or OFF}****>CCENDTLK OFF**

When VersiCOMM is executing a script file, a **Ctrl C** can have differing results depending on whether or not a WAIT command is being executed.

When a **Ctrl C** is keyed during a WAIT, but before the WAIT clears, VersiCOMM aborts the WAIT and executes the next line of the script file, just as if the WAIT had timed out. It also sets a flag that indicates a **Ctrl C** was keyed. Any following CTRLC command will detect that a **Ctrl C** is pending.

When a **Ctrl C** is keyed and a WAIT is not being processed, VersiCOMM aborts the execution of the script file and returns to Conversational mode. Any following CTRLC command is not effective, because the script has been aborted.

CCENDTLK OFF forces COMM to continue to execute the script file when a **Ctrl C** is keyed, and a WAIT is NOT being processed. A flag is also set so that any following CTRLC command detects that a **Ctrl C** has been keyed.

The default is CCENDTLK ON.

**>CSTATUS {ON or OFF}**

If turned OFF this switch suppresses the display of the VersiCOMM command status display. Only the command mode prompt is output.

The default state of this switch is ON.

**>CHGMODE ascii-code or character  
^B****>CHGMODE 2 or >CHGMODE**

Redefines the character used for switching back and forth from Conversational to Command mode. Initially set to Control-^, ASCII 30. (This is the HOME key on many terminals.)

You may specify either the ASCII code for the character, the character itself, or a VersiCOMM flag that contains the ASCII code for the character. Control characters are indicated by a preceding '^'. VersiCOMM flags are indicated by the character string "FLGn," where n can be 0 to 15.

**>CABORT {ON or OFF}****>CABORT ON**

If CABORT is ON, VersiCOMM will exit when DCD (usually pin 8) goes into a low state.

**>CLEAR**

Clears the input buffer of the local communications terminal of any output from the remote computer.

**>CLOSE**

Closes the output SAVE file that is open, if any, so another can be opened. Used to create multiple output files of displayed text that you wish to save.

**>COMMAND ascii-code or character  
^E**

**>COMMAND 5 or >COMMAND**

Redefines the character used for indicating a VersiCOMM command in a USING file. This command would normally only be specified in a USING file. Initially set to the dollar-sign, "\$".

You may specify either the ASCII code for the character, the character itself, or a VersiCOMM flag that contains the ASCII code for the character. Control characters are indicated by a preceding '^'. VersiCOMM flags are indicated by the character string "FLGn," where n can be 0 to 15.


**>COMMENT comment-line**




Inserts the specified comment-line in a SAVE file, if one is currently open.


**>CTRLC label-number**



**>CTRLC 900**

This command checks to see if a Control-C abort is pending and if so, branches forward (only) to the label line with the label-number specified. Would normally only be used in a USING file. The Control-C abort condition is cleared by this command.

When VersiCOMM is executing a script file, a  can have differing results depending on whether or not a WAIT command is being executed.

When a  is keyed during a WAIT, but before the WAIT clears, VersiCOMM aborts the WAIT and executes the next line of the script file, just as if the WAIT had timed out. It also sets a flag that indicates a  was keyed. Any following CTRLC command will detect that a  is pending.

When a  is keyed and a WAIT is not being processed, VersiCOMM aborts the execution of the script file and returns to Conversational mode. Any following CTRLC command is not effective, because the script has been aborted.

NOTE: The CCENDTLK OFF command forces COMM to continue to execute the script file when a  is keyed, and a WAIT is NOT being processed. A flag is also set so that any following CTRLC command detects that a  has been keyed. The default for CCENDTLK is ON.

**>DECFLG #**

**>DECFLG 0**

The DECFLG command is used to decrement the value of one of sixteen VersiCOMM flags. The flags are numbered from 0 to 15, and can take a value from 0 to 255.

The flag number may also be a VersiCOMM flag. For example if flag #1 contains 0 then DECFLG FLG1 is the same as DECFLG 0.

These flags are included in the information that can be exchanged with AlphaBASIC using the VersiCOMM-PLUS interface.

**>DIAL number-string****>DIAL 5551212****>DIAL filespec****>DIAL PHONE.NBR**

This command is used to dial a modem using a .DL0 "dial" file in [7,121].

It will accept either a phone number or the name of a file that contains a phone number.

The DIAL command argument is stored in VersiCOMM parameter @3, and is passed to the DL (dial) file using this parameter.

If it is necessary to wait for secondary dial tones, use a comma to indicate a pause of approximately two seconds. For example:

```
>DIAL 5551212,,,120120
```

If the modem being used does not use the comma as a pause character the PWRITECR command will translate the commas specified to an appropriate number of pause characters for the modem in use, according to the PCHR and PTIME definitions in the DL file.

The "trmdef.DL0" is one of several modem control files that can be generated by selecting "Setup" from the VersiCOMM Dialer screen, and then "Build" from the Setup screen.

If the STATUS switch is ON, the DIAL command displays call progress info on the bottom status line of the terminal, if available.

**>DIALOG {ON or OFF}**

If turned OFF this switch suppresses the display of the dialog that occurs when a USING file is being processed. This switch should be ON when a USING file is being tested. Default value is OFF.

**>DOSCRIPt filespec****>DOSCRIPt TASK1.TLK**

Results in the USING file specified being processed as though it had been specified in the COMM command line. This command can be executed under the control of another USING file, thereby enabling processing of nested USING files.

Same as SENDTLK command.

**>DOTCRT n****>DOTCRT 0**

The DOTCRT command provides a direct method of invoking a terminal TCRT function from a script file, rather than having to use \$AMOS XY = n.

**>DIREXIT ascii-code or character****>DIREXIT 64 or >DIREXIT @**

Redefines the character that enables the user to exit VersiCOMM directly from conversational mode. Setting the ascii-code equal to 255, or any other character that cannot be generated at the keyboard, will disable a direct exit. Initially is set to "~", ASCII 126.

You may specify either the ASCII code for the character, the character itself, or a VersiCOMM flag that contains the ASCII code for the character. Control characters are indicated by a preceding '^'. VersiCOMM flags are indicated by the character string "FLGn," where n can be 0 to 15.

**>DIRSAVE ascii-code or character**

**>DIRSAVE 1 or >DIRSAVE ^A**

Redefines the character used for toggling the "direct" SAVE function. This turns the SAVE switch on and off with leaving Conversational mode. Setting the ascii-code to 255, or any other code that cannot be generated at the keyboard will disable the SAVE feature while in Conversational mode. Initially is set to Control-G, ASCII 7.

You may specify either the ASCII code for the character, the character itself, or a VersiCOMM flag that contains the ASCII code for the character. Control characters are indicated by a preceding '^'. VersiCOMM flags are indicated by the character string "FLGn," where n can be 0 to 15.



EARLIER VERSIONS OF COMM DEFINED 'CONTROL-A' AS THE DEFAULT FOR THIS FUNCTION. THIS DEFINITION INTERFERED WITH THE USE OF FUNCTION KEYS WHEN AN APPLICATION WAS BEING OPERATED REMOTELY ON THOSE TERMINALS THAT USED 'SOH' (CONTROL-A) AS A LEAD-IN CHARACTER FOR FUNCTION KEY SEQUENCES.

IF YOU FIND YOU "CANNOT USE YOUR FUNCTION KEYS" REMOTELY, CHECK AND SEE IF ANY OF THE VERSICOMM CONTROL KEY DEFINITIONS USE A CONTROL CHARACTER THAT IS USED AS THE LEAD-IN CHARACTER IN A FUNCTION KEY SEQUENCE.

**>DISPLAY {ON or OFF}**

If turned OFF this switch suppresses the display of remote characters. If SAVE is currently enabled, incoming characters will still be saved in a disk file.

**>DTR {ON or OFF}**

Toggles the DTR signal "ON or OFF". Requires the use of an Alpha Micro extended function interface driver. If such a driver is not used, "-na-" will be displayed under DTR in the command mode status display.

Alpha Micro extended function interface drivers are included with AMOSL 1.3D and AMOS32 1.0D and later versions of the AMOS operating system.

Not all interfaces allow output signals to be totally independent of system operations. In the case of the AM1000 if RTS (which is used for DTR) is turned OFF, SENDING a file will turn it back on.

**>DTRHANGUP {ON or OFF}**

Toggles the DTRHANGUP function "ON or OFF". When ON, VersiCOMM automatically executes the equivalent of a DTRLOW when a QUIT or Direct Exit is done.



**>DTRLOW {low-time}                      >DTRLOW or >DTRLOW 500**

**>EOLCHR ascii-code or character****>EOLCHR 13 (carriage-return)**

Redefines the character that terminates a line of text in a file that is being transmitted via the SEND or SENDL commands.

When using the SEND command, the end-of-line character determines when VersiCOMM will pause for the amount of time defined by the EOLTIME command.

When using the SENDL command the end-of-character determines when VersiCOMM will wait for the character defined by PROMPTCHR to be transmitted. Initially the end-of-line character is set to be a line-feed.

You may specify either the ASCII code for the character, the character itself, or a VersiCOMM flag that contains the ASCII code for the character. Control characters are indicated by a preceding '^'. VersiCOMM flags are indicated by the character string "FLGn," where n can be 0 to 15.

**>EOLTIME time****>EOLTIME .05**

Sets the interval of time that the SEND command will pause after the end-of-line character has been processed.

If a decimal point is included in the number it is interpreted as floating point seconds to the nearest 0.01 second; otherwise the time specified is considered to be in tenths of a second, i.e. 'EOLTIME 10' would be one second, while 'EOLTIME 10.' would be ten seconds. (This follows a convention established by the PAUSE command and is necessary for compatibility with older TLK files.)

NOTE: Except in special situations, EOLTIME should be set to zero when using the SENDL and SENDEPX commands. If not zero the end-of-line pause will occur before SENDL waits for the next line-prompt character, and after SENDEPX transmits each packet.

**>ESCAPE**

Suspends the operation of VersiCOMM and returns directly to the AMOS monitor without hanging up the modem or processing an OFF file.

PRESERVES THE STATE OF VERSICOMM EXACTLY AS IT IS AT THE TIME THE ESCAPE COMMAND IS EXECUTED. All VersiCOMM switches, variables, and files are preserved. The modem in use by VersiCOMM will remain detached and marked in use. (For example, you will not be able to TRANS a file from AMOS command level after ESCAPEing, since the modem is marked in use.)

To return to VersiCOMM, enter the command "COMM".

Once you have ESCAPed from VersiCOMM you MUST eventually return to VersiCOMM, in order for normal operation of your job and the modem to be restored.

The ESCAPE command cannot be used in a USING file. If VersiCOMM encounters an ESCAPE command in a USING file, it executes an EXIT instead.

An EXIT command should be used in those instances where you wish to exit to AMOS command level without hanging up or processing an OFF file, but do not want to suspend, and then return to, VersiCOMM operation.

### >EXIT

Exit VersiCOMM and return to the AMOS monitor without hanging up the modem or processing an OFF file.

### >FKEY {ON or OFF}

VersiCOMM normally passes sequences generated by function keys to the remote computer, rather than processing the sequence locally. This allows the user to operate a remote application that requires the use of the function keys. When FKEY is ON local translation of function keys is enabled and the AMOS command SET PFK can be used to store sequences for local use in VersiCOMM.



IF YOU FIND YOU "CANNOT USE YOUR FUNCTION KEYS" REMOTELY, CHECK AND SEE IF ANY OF THE VERSICOMM CONTROL KEY DEFINITIONS USE A CONTROL CHARACTER THAT IS USED AS THE LEAD-IN CHARACTER IN A FUNCTION KEY SEQUENCE. SEE THE HELP FILE FOR THE 'DIRSAVE' COMMAND FOR MORE INFORMATION.

### >FLOW {HW or SW} {ON or OFF}

### >FLOW HW ON

This command can be used to control the flow of data transmitted from a device such as a computer or protocol converter to the Alpha Micro. It requires that RCLOCK.SYS be installed in SYSTEM memory and started with RCLOCK following the last SYSTEM command.

Flow control is achieved by monitoring the number of characters in the input buffer and stopping the flow of data into the port when a threshold level is exceeded. Controlling the flow of data may be accomplished via the port hardware using "hardware flow control," or via control characters that are mixed in with other data. This is called "software flow control." The remote device MUST support the type of flow control that is used.

Hardware flow control is the default method. It requires that the transmitting device be controllable via a RS-232 input and that the interface cable connect pin 5 of the Alpha Micro to the controlling pin of the device, usually pin 4.

Software flow control requires that the transmitting device support the XON/XOFF method of flow control.



USE OF THE FLOW COMMAND REQUIRES THAT THE SYSTEM MODULE RCLOCK BE INSTALLED. See the VersiCOMM INSTALLATION Manual for more information about RCLOCK.

If you do not specify HW or SW, HW is assumed.

To minimize the possibility of losing input data, in addition to managing the input buffer it is necessary to prevent "overflow" of the port hardware. This can only be done by the port hardware itself. The FLOW command attempts to enable this feature when it is invoked. Many, but not all, Alpha Micro interfaces have this capability.

For example if an AM1000 with an AM1003 interface was directly connected via an RS-232 port to a central AM2045 and a cable was made in the following manner, FLOW could be used to prevent the AM2045 from overrunning the AM1000 at higher baud rates.

AM1000 (AM1003)		AM2045 (AM355)
port pin 2	<--input--<	port pin 3
port pin 3	>--output->	port pin 2
port pin 4	-<--input--<	CTS pin 5 (output flow control)
port pin 5	>--output->-	RTS pin 4 (input flow control)
port pin 7		port pin 7

#### >FLUSH {LOCAL or REMOTE}

Clears the translation table specified. If none is specified, both the LOCAL and REMOTE tables are cleared.

#### >GOTO label-number

#### >GOTO 900

Processing of the USING file will branch to the label line with the label-number specified. Would normally only be used in a USING file.

#### >HANDSHAKE {ON or OFF}

Enables or disables X/ON, X/OFF protocol when SENDING a sequential file. Default is OFF.

#### >HANGUP

Executes the modem control USING file trmdef.HU0, where "trmdef" is the name of the modem currently in use by VersiCOMM. Note that VersiCOMM would normally do this automatically when a QUIT command or Direct Exit is invoked.

A HANGUP command must never be used in a hangup USING file.

#### >HELP {command}

#### >HELP SEND

Displays the information given in this chapter for the VersiCOMM command specified. If no command is specified a table of VersiCOMM commands is displayed. Requires the VersiCOMM help files in [77,27] (with the extension HCM) to be copied to DSK0:[7,0].

**>INCFLG #****>INCFLG 0**

The INCFLG command is used to increment the value of one of sixteen VersiCOMM flags. The flags are numbered from 0 to 15, and can take a value from 0 to 255.

The flag number may also be a VersiCOMM flag. For example if flag #1 contains 0 then INCFLG FLG1 is the same as INCFLG 0.

These flags are included in the information that can be exchanged with AlphaBASIC using the VersiCOMM-PLUS interface.

**>IDLEOUT {idle-time or ON or OFF}****>IDLEOUT ON or >IDLEOUT 60**

When invoked, IDLEOUT specifies the amount of time that must pass with no activity before VersiCOMM automatically exits. It requires that RCLOCK.SYS be installed in SYSTEM memory and started with RCLOCK following the last SYSTEM command.

The 'idle-time' specified may range in value from 1 to 65535 seconds. If 'ON' is specified instead of an 'idle-time', a default time of 600 seconds is used. If 'OFF' is specified, then no idleout exit can occur.

When the 'idle-time' specified is exhausted, then VersiCOMM exits just as if a user had invoked a direct exit or a carrier abort exit had occurred.

IDLEOUT has no effect when VersiCOMM is running under the control of a USING file.

If desired, an IDLEOUT command can be placed in the INI.TLK file to enable this feature whenever VersiCOMM is executed.

**>INPARG #****>INPARG 0**

VersiCOMM has the capability to pass up to three 30 character strings and 13 60 character strings from the VersiCOMM command line to commands in a USING file. For example,

```
.COMM USING TALK WITH FILE1.LST FILE2.LST FILE3.LST 555-1212
```

would make available the strings FILE1.LST, FILE2.LST, FILE3.LST and 555-1212 to a VersiCOMM USING file. You would specify these strings using the "tokens" @0, @1, @2, and @3 in their place in the USING file. For example,

```
$LOOKUP @0
$GOTO 100
$TRANS @0
```

The @3 token is reserved for use by the DIAL command. Tokens greater than @9 are referred to as @A, @B, @C, @D, @E and @F.

The INPARG command is used to input the value of these tokens from within the USING file, instead of setting them from the command line.

The following segment of USING file code would set the value of @0 to a user provided password:

```
$TYPE Password
$INPARG 0
```

These arguments are included in the information that can be exchanged with AlphaBASIC using the VersiCOMM-PLUS interface.

#### **>INPINC {ON or OFF}**

Can be used to set or reset the VersiCOMM internal INC.FL. This flag is used to specify whether or not a special input routine (used for flow control) should be used.

#### **>INUSE {ON or OFF}**

Turns the communications port "inuse" flag ON or OFF. This command is only used in special cases. The "inuse" flag is normally set or cleared automatically.

#### **>JWAIT {ON or OFF}**

Turning JWAIT ON results in VersiCOMM operating in such a manner that no CPU time is used unless there is local or remote activity. A '\$JWAIT ON' command can be added to the INI.TLK file if you wish this to be the default mode of operation.

Requires that the VersiCOMM interval timer RCLOCK be implemented. See the VersiCOMM INSTALLATION MANUAL for more information.

#### **>LABELCHR ascii-code or character !**

**>LABELCHR 33 or >LABELCHR**

Redefines the character used to indicate a "label line." This is a line to which a GOTO can branch. Initially is set to a semi-colon.

You may specify either the ASCII code for the character, the character itself, or a VersiCOMM flag that contains the ASCII code for the character. Control characters are indicated by a preceding '^'. VersiCOMM flags are indicated by the character string "FLGn," where n can be 0 to 15.

#### **>LINEFEED {ON or OFF}**

Enables or disables the sending of linefeeds, when SENDING a sequential file. Default is ON, that is linefeeds are sent.

#### **>LISTEN {number}**

**>LISTEN 10 (in seconds)**

Used in USING files to transfer to Conversational mode and capture incoming data that has resulted from issuing a remote computer command. Returns to Command mode if no characters are received from the remote computer for the number of seconds specified in the argument so processing of a USING file may continue. If no argument is specified, the value defaults to about 30 seconds.

#### **>LOCAL code1,code2**

**>LOCAL 65,66 ("A" to "B")**

Adds code1 to the local translation table. This will result in local code1 being converted to (remote) code2 before sending it to the remote computer. The code specified is the decimal

value of the character code. If code2=0 local code1 is "filtered" and not sent to the remote computer. If code2 is not specified, the current translation of code1 is returned. If code2=code1 the entry is removed from the translation table and the character will not be translated or filtered. For example,

```
LOCAL 65,193
```

will convert any ASCII "A" you type into a EBCDIC "A" before sending it to the remote computer. And,

```
LOCAL 65,0
```

will discard any ASCII "A" you type instead of sending it to the remote computer. And,

```
LOCAL 65
```

will tell you what code "A" is converted to. And,

```
LOCAL 65,65
```

removes the entry for character 65 from the translation table.

### **>LOOKUP filespec {size}**

### **>LOOKUP FILE.LST 5**

The LOOKUP command provides a means for conditional branching in a USING file based on the existence or (optionally) the size of a file.

Filespec is a sequential or random file specification and size is an optional parameter specifying how large the file must be, in bytes, to "pass" the lookup.

If the LOOKUP passes then the line immediately following the LOOKUP command line in the USING file is skipped, otherwise if the LOOKUP fails the line is executed.

### **>MODEM # [p,pn]**

### **>MODEM 2 [7,122]**

This command assigns a default [p,pn] to a USING file filespec conditionally based on the modem COMM selects. "#" is the sequence number of the modem and [p,pn] is the account number that will be used if a PPN is not specified for a USING file in a COMM command line. For example, if the command line COMM USING MCI were entered, the following INI.TLK file would result in COMM looking in [77,31] for MCI if it is using MODEM1, but [77,32] if it is using MODEM2:

```
$MODEM 1 [77,31]
$MODEM 2 [77,32]
```

This command should only be used in an "INI.TLK" file.

**IMPORTANT**--if a [p,pn] is explicitly specified for a USING file in the COMM command line, MODEM commands will be ignored.

**>MOVCUR y x****>MOVCUR 22 1**

The MOVCUR command provides a direct method of performing terminal cursor positioning from a script file, rather than having to use \$AMOS XY y x.

The cursor is moved to column x of row y.

**>MVER character**

MVER can be used to specify an alternate attention, dial, or hang-up file. Each of these files have extensions that normally end in "0", i.e. MODEM1.AT0, MODEM1.DL0, and MODEM1.HU0. The MVER command can be used to specify that an alternate character be used for the last character of the extension. For example the command,

**>MVER 1**

would result in MODEM1.AT1, MODEM1.DL1, and MODEM1.HU1 being used.

**>NEWLINE {ON or OFF}**

NEWLINE is used with the VT100 emulation to set NEWLINE mode ON or OFF. If ON, a carriage-return received from the host will be interpreted as a carriage-return followed by a line-feed.

If NEWLINE is set ON, the AUTOLF switch should also be set ON so that keying a carriage-return generates a carriage-return and a line-feed.

**>NOTELOG comment-line**

Inserts the specified comment-line in the VersiCOMM LOG file, if it has been activated. VersiCOMM will maintain a log file if the file trmdef.CLG exists in the account from which VersiCOMM is being executed.

**>NOTESAVE comment-line**

Inserts the specified comment-line in a SAVE file, if one is currently open. Same as COMMENT.

**>ORFLG n m****>ORFLG 0 5**

The ORFLG command is used to logical "or" a value m to the value contained in the VersiCOMM flag n. The value, m, may range in value from 0 to 255. The flag number n may be 0 to 15.

Both n and m may also be VersiCOMM flags. For example if flag #1 contains 0 and flag #2 contains 5, then ORFLG FLG1 FLG2 is the same as ORFLG 0 5.

These flags are included in the information that can be exchanged with AlphaBASIC using the VersiCOMM-PLUS interface.



**>PARAMCHR ascii-code or character**  
%

**>PARAMCHR 37 or >PARAMCHR**

VersiCOMM has the capability to pass up to three 30 character strings and 13 60 character strings from the VersiCOMM command line to commands in a USING file. For example,

```
.COMM USING TALK WITH FILE1.LST FILE2.LST FILE3.LST 555-1212
```

would make available the strings FILE1.LST, FILE2.LST, FILE3.LST and 555-1212 to a VersiCOMM USING file. You would specify these strings using the "tokens" @0, @1, @2, and @3 in their place in the USING file. For example,

```
$LOOKUP @0  
$GOTO 100  
$TRANS @0
```

The @3 token is reserved for use by the DIAL command. Tokens greater than @9 are referred to as @A, @B, @C, @D, @E and @F.

The PARAMCHR command redefines the character used as a lead-in character for token substitution. Initially this character is an "@".

You may specify either the ASCII code for the character, the character itself, or a VersiCOMM flag that contains the ASCII code for the character. Control characters are indicated by a preceding '^'. VersiCOMM flags are indicated by the character string "FLGn," where n can be 0 to 15.

**>PARITY {ODD, EVEN or NONE}**

**>PARITY EVEN**

Changes the parity of the port VersiCOMM is currently using for communications.

This command requires the use of an Alpha Micro extended function interface driver. Extended function interface drivers are included with AMOSL 1.3D and AMOS32 1.0D and later versions of the AMOS operating system. If such a driver is not used, "-na-" will be displayed under PARITY in the command mode status display.

Not all combinations of parity, word size and number of stop bits are valid. For example, on an AM1000 port you cannot specify PARITY as 'NONE' when the word size is 7. You must first use the SIZE command to change the word size to 8, and then change the PARITY to NONE.

Also see the SPARITY command. SPARITY enables you to use 7 data bits and either EVEN, ODD, or STRIP parity without the need for a special interface driver.

**>PAUSE number**

**>PAUSE 50 or >PAUSE 5.0**

Results in COMM pausing for the specified number of seconds. If 'number' contains no decimal point, its value is interpreted as being in tenths of a second, i.e. '50' is interpreted as five seconds. If a decimal point is included, the number is used as specified.

This command would normally only appear in a USING file.

**>PCHR character**

PCHR is a specialized command that is normally only use in a dial modem control module. It defines the pause character for the modem being dialed.

**>PRESERVE {ON or OFF}**

If turned ON VersiCOMM will preserve the contents of the screen when switching between Conversational and Command modes.

To use PRESERVE your terminal MUST be an AM65A or other terminal that supports the "Save Screen Area" function.

**>PRINT comment-line**

Displays the specified comment-line on the user's screen during the execution of a USING file. Normally would only be used in a USING file.

**>PROMPTCHR ascii-code or character****>PROMPTCHR 46 or****>PROMPTCHR .**

Redefines the character that the SENDL command must receive after it has processed the currently defined end-of-line character, before it will start to process the characters in the next line of text.

You may specify either the ASCII code for the character, the character itself, or a VersiCOMM flag that contains the ASCII code for the character. Control characters are indicated by a preceding '^'. VersiCOMM flags are indicated by the character string "FLGn," where n can be 0 to 15.

If just 'PROMPTCHR' is entered without an argument, then the current prompt character is returned. Initially the prompt character is set to be a colon.

**>PTIME time**

PTIME is a specialized command that is normally only used in a dial modem control module. It defines the time interval for a pause character for the modem being dialed.

**>PWRITECR string**

PWRITECR is a specialized version of the WRITECR command that is normally only used in a dial modem control module.

**>QUIT {filespec}****>QUIT**

Exit VersiCOMM and return to the AMOS monitor. If a filespec is specified it processes this USING file before exiting. If no filespec is specified then IF a USING file was processed when COMM was first executed, VersiCOMM checks if there is an .OFF file with the same name (and account) as that first USING file. If it finds one it processes it.

AFTER processing any .OFF file, VersiCOMM does a similar check for a trmdef.HU0 (or HANGUP.TLK) file, where "trmdef" is the AMOS TRMDEF of the modem terminal. If such a file is found it is processed.

Use the >EXIT command to exit VersiCOMM without checking for .OFF or .HUx files.



**DO NOT USE THE QUIT COMMAND IN A trmdef.HU0 or HANGUP.TLK FILE!**

#### **>RECEIVE filespec**

#### **>RECEIVE ORDER.RUN**

This command provides an alternative to using the SAVE command for saving file structured data that is sent from a remote computer. After this command is executed all data received by the communications port will be stored in the file specified. An 'end of file' is interpreted to be a pause of more than 1 second in the data stream. RECEIVE will wait for about 1 minute for the initial data to arrive. TRANSLATION OF CHARACTERS WILL BE PERFORMED IF REMOTE TRANSLATION IS ENABLED. THE HIGH BIT OF EACH CHARACTER WILL BE ZEROED IF "STRIP" IS SET "ON".

#### **>RECORD {ON or OFF}**

#### **>RECORD**

The RECORD command is used in combination with the SAVE command, when an emulation is in effect, to produce recordings of output from remote systems that can be played back in simulated real-time using a player written for the emulation.

This feature can be used to record demonstrations that can be played back on other systems.

For example, you could RECORD an Alpha Micro session using the GENeric emulation and then play it back using the player SHWGEN.

Note that SAVE must be ON for RECORD to be effective, and that an emulation such as GEN or VT1 must be enabled if the resulting file is to be played back at a later time.

#### **>RECX filespec**

#### **>RECX TEST.COM**

RECX is an XMODEM option function that will receive a file from a PC that supports the XMODEM protocol. Both CRC and checksum error checking is supported.



**BEFORE RUNNING RCX YOU MUST FIRST INITIATE THE XMODEM SEND FUNCTION ON THE REMOTE PC.**

RECX can be run at AMOS command level or directly from VersiCOMM.

>RECX/switch >rcxz/P

RECX receives sequential files using the ZMODEM or YMODEM protocol. ZMODEM is generally regarded a much more robust protocol and if the sending system supports both protocols ZMODEM should normally be selected.

For RCX to properly execute, RCLOCK must be installed. Unless it is run from a remote system using the /REMOTE option, RCX must be run from VersiCOMM command mode.

RECZ uses the **FastQ** enhanced serial communications system if the file RECZ.INI exists (on the local system) in the account from which RECZ is being run, or if RECZ.INI exists in DSK0:[7,0]. At this time it is not necessary to place a FASTQ ON command in this file. If the file RECZ.INI is not found, **FastQ** is not used.

Beginning with version 1.1(102), RECZ follows AMOS rather than UNIX conventions. Switches must be preceded by "/".

No file names should be specified in the RECZ command line.

You should initiate the ZMODEM or YMODEM transmit program on the remote system prior to entering the RECZ or RECZ/YMODEM command at the VersiCOMM command mode prompt. For example, when using AMTEC+ you must specify that you wish to download files, the file(s) you want to download, and that you will be using the ZMODEM protocol.



If auto-downloading is enabled, it will not be necessary to switch to command mode and enter the RECZ command.

RECZ supports the resumption of an interrupted file transfer session. If a "download resume" command is available on the remote system it is not necessary to explicitly specify the "/R" switch in the RECZ command line

The following switches are defined:

/APPEND	Append to existing destination file
/BINARY	Binary file transfer (default)
/DELETE	Delete existing file if any
/ESCAPE	Escape control characters
/FLOW	Manage input buffer using RTS flow control (default)
/PARTIAL	Allow reception of partial file so can use RESUME (default)
/PROTECT	Protect existing destination file
/REMOTE	Enable use at AMOS command level as a remote command
/RESUME	Resume interrupted file transfer
/RETRY:n	Change the maximum number of retries to n
/SEGMENT	Use segmented streaming (default on 68000 and 68010 systems)
/TIMEOUT:n	Change timeout to n tenths of sec
/WINDOW:n	Change window to n characters
/YMODEM	Use YMODEM batch protocol

**>REDIAL**

This command will redial the number specified in the last DIAL command processed.

**>REMOTE code1,code2****>REMOTE 66,65 ("B" to "A")**

Adds code1 to the remote translation table. This will result in remote code1 being converted to (local) code2 upon receipt from the remote computer. If code2=0, remote code1 is filtered and not accepted locally. If code2 is not specified, the current translation of code1 is returned. If code2=code1 the entry is removed from the translation table and the character will not be translated or filtered.

**>RETRV {local-fspec=} remote-fspec****>RETRV \*.NEW=\*.LIT**

RETRV will retrieve ANY type of AMOS file with CRC16 block error correction and data compression from another Alpha that is equipped with the companion program TNZA. TNZA is included on AMOS/L 1.3B and AMOS32 1.0A and later operating systems, as well as with VersiCOMM-Plus.

Once a link has been established using VersiCOMM, RETRV can be run at AMOS command level or directly from VersiCOMM-Plus.

RETRV accepts wildcards and output filespecs similar in operation to the AMOS COPY command. The MEM: device may be specified in input or output file specifications. A device that is not defined on the local system may be specified as an output device.

If adequate memory is available on the *remote* system, RETRV uses ***FasTrans*** data compression as a default. This type of compression normally reduces the number of characters transferred by 45% to 75% or more. If insufficient memory is available to use ***FasTrans*** compression, then Adaptive compression is used.

RETRV uses the ***FastQ*** enhanced serial communications system if the file RETRV.INI exists (on the local system) in the account from which RETRV is being run, or if RETRV.INI exists in DSK0:[7,0]. At this time it is not necessary to place a FASTQ ON command in this file. If the file RETRV.INI is not found, ***FastQ*** is not used.

If you want RETRV to "overwrite" an existing file on the local computer you may use either the /DELETE, /UPDATE, /HASH, or /VERSION switches. The /DELETE switch deletes the file unconditionally. The /UPDATE or /HASH switches are the same, and delete the file only if it is different from the file on the remote system. The /VERSION switch deletes the file if its version number is earlier than the file that would replace it.

If the /L switch is specified, RETRV will log file transfers in the file RETRV.LOG.

If needed, RETRV will create an account on the local system if the user is logged into the operator account of the local system.

RETRV configures itself depending upon whether the remote system is a "slow" or fast" system and a high-speed transfer is being requested.

For more information about RETRV see the Chapter "Sending and Receiving Files."

**>RTS {ON or OFF}**

Toggles the RTS signal "ON or OFF". Requires the use of an Alpha Micro extended function interface driver. If such a driver is not used, "-na-" will be displayed under RTS in the command mode status display.

Alpha Micro extended function interface drivers are included with AMOSL 1.3D and AMOS32 1.0D and later versions of the AMOS operating system.

**>RTSLOW {low-time}****>RTSLOW or >RTSLOW 500**

Drops the 'RTS' line to the modem. The line will remain low for the currently defined low-time. Initially this is 250 milliseconds. This default can be changed by specifying an argument in the command. The argument is the new low-time in milliseconds.

The low-time argument may be specified in a VersiCOMM flag. When used as an argument in a command a VersiCOMM flag is written as "FLGn", where n is the flag number, e.g. FLG0, FLG1, ..., FLG15.

If RTS is not present on the interface and DTR is available, DTR will be dropped low instead.

This command requires the use of an Alpha Micro extended function interface driver. Extended function interface drivers are included with AMOSL 1.3D and AMOS32 1.0D and later versions of the AMOS operating system.

**>RUN basic-run-file****>RUN PRGRM**

The VersiCOMM 'RUN' command is used to run a BASIC program in a USING file or from VersiCOMM command level.

NOTE: Be sure to use only RUN files that are compatible with the AlphaBASIC software on your computer, i.e. don't try to RUN files compiled with the "old" compiler if you are on a system that uses the "new" compiler.

If you are using VersiCOMM-PLUS, RUN also provides a means for passing information from the VersiCOMM command line to AlphaBASIC, exchanging information between USING files and AlphaBASIC, and enabling the use of the TREAD and TWRITE subroutines for communicating with the modem port from AlphaBASIC.

The following XCALLs are used to pass information to the BASIC program and then back to VersiCOMM.

XCALL GCMARG,ARGBLK,TCODE (Gets the data from VersiCOMM)

XCALL PCMARG,ARGBLK,TCODE (Puts the data back in VersiCOMM)

GCMARG would normally be called at the logical beginning of the BASIC program, and PCMARG at the logical end.

See the Chapter "VersiCOMM-Plus Basic Interface" for more important information.

**>SAVE {filespec or ON or OFF}****>SAVE OUTPUT.LST**

If 'filespec' is specified, all displayed data is copied to the file specified. If an output file is already opened, an error will result. A default extension of LST is assumed. "ON.LST" and "OFF.LST" will not be recognized as SAVE filespecs.

If 'ON' is specified and an output file has already been opened, the additional text is appended to the file.












If 'ON' is specified and no output file has been opened, the file "terminal-name.LST" is used for an output file and all subsequent displayed data will be copied to this file.

If 'OFF' is specified, copying of displayed data will be suspended.

**>SELECT filespec****>SELECT \*.BAS**

Invokes the *VersiCOMMander* file command executive. *VersiCOMMander* is a point and select file manager that uses selections from pull down menus to perform various file functions. This includes sending and receiving files using several protocols, viewing files, calculating hash totals, copying files, renaming files, and erasing files.

Following is a summary of the keyboard controls used by *Versi COMMander*.

<SPACE>	Select the file at the current cursor position.
<ESC>	Display or remove the active sub-menu.
	Display the File sub-menu.
	Display the Send sub-menu.
	Display the Receive sub-menu.
	Display the Settings sub-menu.
	Display the System sub-menu.
Cursor Keys	Position the file or sub-menu cursor.
	Perform the action specified in the displayed sub-menu.
	Return to VersiCOMM command mode.
	View the file at the cursor.
	Enter a new file set specification.
	Enter a LIT command to apply to currently selected files.
	Calculate the hash total of the currently selected files.

**>SEND filespec {percentage}****>SEND TEXT.TXT 50**

Sends the sequential file specified to the remote computer. Linefeeds will be suppressed or sent, depending on the state of the LINEFEED switch.

TRANSLATION OF CHARACTERS WILL BE PERFORMED IF LOCAL TRANSLATION IS ENABLED. An optional throttle percentage may be specified in order to slow the speed of the transfer by the percentage indicated. In the command example a throttle percentage of 50% has been specified. The maximum percentage that may be specified is 65535%. Using a throttle percentage of about 4000 a file can be "sent" to VUE running in text entry mode on a remote computer.

If the currently defined end-of-line time is non-zero, SEND will pause for the end-of-line time when the end-of-line character has been processed.

#### **>SENDEPX filespec {packet-size}**

Sends the sequential file specified to the remote computer with "echoplex" error checking. If an error is detected the packet transmitted indicator, "-", changes to an asterisk. When the transmission of the file has been completed, if no errors were detected SENDEPX returns to conversational mode; otherwise you are notified of a possible error and SENDEPX returns to the command mode prompt.

For SENDEPX to work properly the remote computer MUST echo each character it receives. NO conversions, such as converting a TAB to spaces, may take place. The VersiCOMM LINEFEED switch MUST be OFF if sending a file that contains line-feeds. In addition to any filters you have specified, SENDEPX will filter any linefeeds or nulls it receives from the remote computer. This permits an echo of <CR><LF>, optionally followed by nulls, when a <CR> has been transmitted.

SENDEPX transmits a series of character packets. The number of characters in a packet may be optionally specified in a parameter following the file specification. The maximum (and default) packet size is 64, the minimum size is 1. The smaller the packet size, the slower the rate of the transfer.

The HANDSHAKE switch has no effect on the operation of SENDEPX. If the EOLTIME value is non-zero, SENDEPX will pause for this interval after each packet is transmitted.

#### **>SENDL filespec {percentage}**

#### **>SENDL TEXT.TXT 50**

This command is intended to be used for sending text to a line editor that prompts the user for each line of text. For example:

```
1:This is the first line of text.  
2:
```

Before sending the file the prompt character must be defined using the PROMPTCHR command and the end-of-line character must be defined using the EOLCHR command. The default prompt character is a colon. If PROMPTCHR is set to a null (ASCII 0), SENDL does not wait for a prompt before sending the next line. (The end-of-line time should normally be zero when sending text in this manner.)

Linefeeds will be suppressed or sent, depending on the state of the LINEFEED switch. Normally you would suppress linefeeds when using SENDL.

TRANSLATION OF CHARACTERS WILL BE PERFORMED IF LOCAL TRANSLATION IS ENABLED. An optional throttle percentage may be specified in order to slow the speed



of the transfer by the percentage indicated. In the command example a throttle percentage of 50% has been specified. The maximum percentage that may be specified is 65535%.

**>SENDTLK filespec**

**>SENDTLK TASK1.TLK**

Results in the USING file specified being processed as though it had been specified in the COMM command line. This command can be executed under the control of another USING file, thereby enabling processing of nested USING files.

**>SENDX filespec**

**>SENDX FILE.LST**

SENDX is an XMODEM option function that will transmit a file to a PC that supports the XMODEM protocol. Both CRC and checksum error checking is supported.



**BEFORE RUNNING SENDX YOU MUST FIRST INITIATE THE XMODEM RECEIVE FUNCTION ON THE REMOTE PC.**

SENDX can be run at AMOS command level or directly from VersiCOMM.

**>SENDZ fspec1,fspec2,...fspecn /switch**

**>sendz \*.txt/D**

SENDZ transmits sequential files using the ZMODEM or YMODEM protocol. ZMODEM is generally regarded a much more robust protocol and if the receiving system supports both protocols ZMODEM should normally be selected.

For SENDZ to properly execute, RCLOCK must be installed. Unless it is run from a remote system using the /REMOTE option, SENDZ must be run from VersiCOMM command mode.

Beginning with version 1.1(102), SENDZ supports wildcarding and follows AMOS, rather than UNIX, command line conventions.

When using the /YMODEM option you must first start the YMODEM receive program on the remote system, prior to entering the SENDZ/YMODEM command at the VersiCOMM command mode prompt. In some cases this may also be necessary when using the ZMODEM protocol. For example, when using AMTEC+ you must specify that you wish to upload files and that you will be using the ZMODEM protocol. SENDZ transmits sequential files using the ZMODEM protocol. The receiving system must support the ZMODEM protocol.

The following switches are defined:

/APPEND	Append to existing destination file.
/ASCII	Change NL to CR/LF.
/BINARY	Binary file transfer override (default).
/BLOCK:n	Limit subpacket length to n bytes.
/DELETE	Delete existing file, if any.
/ESCAPE	Escape all control characters.
/FRAME:n	Limit frame length to n bytes.

/FULL	Send Full pathname.
/LONGER	Send file if source longer.
/OLDCRC	Use 16 bit CRC instead of 32 bit CRC.
/ONEK	Send 1024 byte subpackets.
/PROTECT	Protect existing destination file.
/REMOTE	Enable use at AMOS command level as a remote command
/REPLACE	Replace existing file, skip if not present at remote.
/RESUME	Resume/Recover interrupted file transfer.
/SLASH	Change '.' to '/' in pathnames.
/TIMEOUT:n	Change timeout to n tenths of a sec.
/WINDOW:n	Window is n bytes.
/YMODEM	Use YMODEM batch protocol

**>SETARG # string****>SETARG 0 FILE.LST**

VersiCOMM has the capability to pass up to three 30 character strings and 13 60 character strings from the VersiCOMM command line to commands in a USING file. For example,

```
.COMM USING TALK WITH FILE1.LST FILE2.LST FILE3.LST 555-1212
```

would make available the strings FILE1.LST, FILE2.LST, FILE3.LST and 555-1212 to a VersiCOMM USING file. You would specify these strings using the "tokens" @0, @1, @2, and @3 in their place in the USING file. For example,

```
$LOOKUP @0
$GOTO 100
$TRANS @0
```

The @3 token is reserved for use by the DIAL command. Tokens greater than @9 are referred to as @A, @B, @C, @D, @E and @F.

The SETARG command is used to set the value of these tokens from within the USING file, instead of from the command line. The value of the strings can also be set using the transient INIARG.TLK file.

These arguments are included in the information that can be exchanged with AlphaBASIC using the VersiCOMM-PLUS interface.

**>SETFLG n m****>SETFLG 0 1**

The SETFLG command is used to set the value of one of sixteen VersiCOMM flags. The flag number n can be 0 to 15, and can take a value, m, from 0 to 255.

Both n and m may also be VersiCOMM flags. For example if flag #1 contains 0 and flag #2 contains 5, then SETFLG FLG1 FLG2 is the same as SETFLG 0 5.

These flags are included in the information that can be exchanged with AlphaBASIC using the VersiCOMM-Plus interface.

### **>SIZE number**

### **>SIZE 7**

Changes the number of "data bits" in a "character." SIZE normally does not include the parity bit.

This command requires the use of an Alpha Micro extended function interface driver. Extended function interface drivers are included with AMOSL 1.3D and AMOS32 1.0D and later versions of the AMOS operating system. If such a driver is not used, "-na-" will be displayed under SIZE in the command mode status display.

Not all combinations of parity, word size and number of stop bits are valid. For example, on an AM1000 port you cannot specify SIZE as 7 when the parity is set as 'NONE'. You must first use the PARITY command to change the parity to EVEN or ODD and then change the SIZE to 7.

### **>SPARITY {ODD, EVEN, STRIP or NONE}**

Rather than using the hardware interface to create parity, SPARITY uses software to modify an eight bit code to seven bits plus a parity bit. To use SPARITY the port hardware MUST be set to eight bits and no parity.

For example, the command SPARITY EVEN has, in effect, the same result as the hardware oriented commands SIZE 7, PARITY EVEN, and STOP 1, but DOES NOT require the use of an extended function interface driver.

### **>STATUS {ON or OFF}**

If turned OFF this switch suppresses the display of the VersiCOMM (bottom) status line.

The default state of this switch is determined by the capabilities of the user's terminal. If the user's terminal has a bottom status line, STATUS is turned ON, otherwise it is turned OFF.

The state of this switch can be determined in a script file by using the TRMFLG and TSTFLG commands.

### **>STOP number**

### **>STOP 1**

Changes the number of "stop bits" in a "character."

This command requires the use of an Alpha Micro extended function interface driver. Extended function interface drivers are included with AMOSL 1.3D and AMOS32 1.0D and later versions of the AMOS operating system. If such a driver is not used, "-na-" will be displayed under STOP in the command mode status display.

Not all combinations of parity, word size and number of stop bits are valid. For example, you cannot change the number of stop bits to two when parity is EVEN and SIZE is 8.

**>STRIP {ON or OFF}**

Strips incoming remote characters to ASCII--i.e. masks out the high order bit of the byte. This occurs AFTER translation, if translation is enabled.

Also strips local characters from the keyboard. If no argument is specified, ON is assumed. Initially is set OFF.



When STRIP is OFF, all eight bits of a received byte are used in commands such as the WAIT command. This can lead to situations where a character with the high bit set DOES NOT MATCH a character with the high bit clear, but on the screen both characters look the same!

When using an 8-bit character set on the remote system, STRIP must be turned OFF.

**>SUBFLG n m****>SUBFLG 0 5**

The SUBFLG command is used to subtract a value m from the value contained in the VersiCOMM flag n. The value to be subtracted, m, may range in value from 0 to 255. The flag number n may be 0 to 15.

Both n and m may also be VersiCOMM flags. For example if flag #1 contains 0 and flag #2 contains 5, then SUBFLG FLG1 FLG2 is the same as SUBFLG 0 5.

These flags are included in the information that can be exchanged with AlphaBASIC using the VersiCOMM-PLUS interface.

**>TALKTO**

Normally used in a USING file to end a particular logical sequence in which you wish to leave the user in VersiCOMM conversational mode.

**>TALL {ON or OFF}**

Turns translation of both local and remote characters ON or OFF. If no argument is specified, ON is assumed. Initially is set OFF.

**>TLOCAL {ON or OFF}**

Turns translation of local characters on or off. If no argument is specified, ON is assumed. Initially is set OFF.

**>TRANS {remote-fspec=} local-fspec****>TRANS \*.NEW=\*.LIT**

TRANS will transmit ANY type of AMOS file with CRC16 block error correction and data compression to another Alpha that is equipped with the companion program RZVA. RZVA is included on AMOS/L 1.3B and AMOS32 1.0A and later operating systems, as well as with VersiCOMM-Plus.

Once a link has been established using VersiCOMM, TRANS can be run at AMOS command level or directly from VersiCOMM.

TRANS accepts wildcards and output filespecs similar in operation to the AMOS COPY command. The MEM: device may be specified in input or output file specifications. A device that is not defined on the local system may be specified as an output device.

If adequate memory is available on the *local* system, TRANS uses ***FasTrans*** data compression as a default. This type of compression normally reduces the number of characters transferred by 45% to 75% or more. If insufficient memory is available to use ***FasTrans*** compression, then Adaptive compression is used.

TRANS uses the ***FastQ*** enhanced serial communications system if the file RZVA.INI exists (on the remote system) in the account from which RZVA is being run, or if RZVA.INI exists in DSK0:[7,0] of the remote system.

If you want TRANS to "overwrite" an existing file on the remote computer you may use either the /DELETE, /UPDATE, /HASH or /VERSION switches. The /DELETE switch deletes the file unconditionally. The /UPDATE and /HASH switches are the same, and delete the file only if it is different from the file on the local system. The /VERSION switch deletes the file if the file that would replace the deleted file has a later version number.

If the /L switch is specified, TRANS will log file transfers in the file TRANS.LOG.

If needed, TRANS will create an account on a remote system if the user is logged into the operator account of the remote system.

TRANS configures itself depending upon whether the remote system is a "slow" or fast" system and a high-speed transfer is being requested.

If desired, the filespec part of the command line may be indirectly specified in a file. For example, if the file "FILES.IND" contained the following lines:

```
FILE1.DAT,
FILE2.DAT,
FILE3.DAT
```

The command TRANS @FILES would be equivalent to TRANS FILE1.DAT,FILE2.DAT, FILE3.DAT.

For more information about TRANS see the Chapter "Sending and Receiving Files."

### >TREMOTE {ON or OFF}

Turns translation of remote characters on or off. If no argument is specified, ON is assumed. Initially is set OFF.

### >TRMFLG n

### >TRMFLG 0

The TRMFLG command returns a subset of terminal characteristics in the specified VersiCOMM flag. The flag number n can be 0 to 15. The returned value is comprised of the following values:

- 1 Bottom status line is supported and enabled in VersiCOMM
- 2 Box commands are supported

#### 4 Terminal is a "mode" terminal

So, for example, if the bottom status line is supported and enabled and box commands are supported, the returned value would be 3.

The flag number may also be a VersiCOMM flag. For example if flag #1 contains 0 then TRMFLG FLG1 is the same as TRMFLG 0.

These flags are included in the information that can be exchanged with AlphaBASIC using the VersiCOMM-Plus interface.

#### >TSTARG # string

#### >TSTARG 0 FILE.LST

VersiCOMM has the capability to pass up to three 30 character strings and 13 60 character strings from the VersiCOMM command line to commands in a USING file. For example,

```
.COMM USING TALK WITH FILE1.LST FILE2.LST FILE3.LST 555-1212
```

would make available the strings FILE1.LST, FILE2.LST, FILE3.LST and 555-1212 to a VersiCOMM USING file. You would specify these strings using the "tokens" @0, @1, @2, and @3 in their place in the USING file. For example,

```
$LOOKUP @0
$GOTO 100
$TRANS @0
```

The @3 token is reserved for use by the DIAL command. Tokens greater than @9 are referred to as @A, @B, @C, @D, @E and @F.

The TSTARG command is used to compare the value of a specified argument to a string. If the value of the argument is equal to the string the next line of the USING file is skipped, otherwise it is processed.

The value of the argument can be passed via the VersiCOMM command line, set via the SETARG command, input via the INPARG command or initialized using the transient INIARG.TLK file.

These arguments are included in the information that can be exchanged with AlphaBASIC using the VersiCOMM-PLUS interface.

#### >TSTDIALOG

The TSTDIALOG command is used to test the state of the VersiCOMM DIALOG switch.

This command can only be specified in a USING file. If the current value of the DIALOG switch is ON, the next line of the USING file is skipped, otherwise it is processed.

#### >TSTFLG n m

#### >TSTFLG 0 1

The TSTFLG command is used to test the value of one of sixteen VersiCOMM flags. The flag number n can be 0 to 15, and can take a value, m, from 0 to 255.

This command can only be specified in a USING file. If the current value of the flag is equal to the value specified, the next line of the USING file is skipped, otherwise it is processed.

Both n and m may also be VersiCOMM flags. For example if flag #1 contains 0 and flag #2 contains 5, then TSTFLG FLG1 FLG2 is the same as TSTFLG 0 5.

These flags are included in the information that can be exchanged with AlphaBASIC using the VersiCOMM-Plus interface.

#### **>TYPE comment-line**

Displays the specified comment-line on the user's screen during the execution of a USING file, without a terminating carriage-return, line-feed.

Would normally only be used in a USING file.

#### **>TYPECR comment-line**

Displays the specified comment-line on the user's screen during the execution of a USING file, with a terminating carriage-return, line-feed.

Would normally only be used in a USING file.

#### **>TYPEFLG n**

#### **>TYPEFLG 0**

The TYPEFLG command displays the value of the specified VersiCOMM flag. The flag number n can be 0 to 15. No carriage-return or line-feed is output after the value of the flag is displayed.

The flag number may also be a VersiCOMM flag. For example if flag #1 contains 0 then TYPEFLG FLG1 is the same as TYPEFLG 0.

These flags are included in the information that can be exchanged with AlphaBASIC using the VersiCOMM-Plus interface.

#### **>TYPEWAIT**

#### **>TYPEWAIT**

The TYPEWAIT command displays the current WAITIME value. No carriage-return or line-feed is output after the value is displayed.

TYPEWAIT is normally only be used in a script file.

These flags are included in the information that can be exchanged with AlphaBASIC using the VersiCOMM-Plus interface.

#### **>WAIT wait-string**

#### **USING FILES ONLY**


The \$WAIT command indicates that the characters that follow, up to but not including the carriage-return and line-feed that ends the line, are "wait characters" and are to be received AS A GROUP from the remote computer before processing of the USING file is continued. ONE space should separate the \$WAIT command and the wait string in the command line. Additional leading spaces will be considered significant and included in the wait string.




The interval of time that VersiCOMM will wait for the specified string is time specified in the last \$WAITIME command that was processed by the USING file. If no \$WAITIME command has been executed the wait time defaults to about 40 seconds.


IF VERSICOMM RECEIVES THE SPECIFIED STRING IN THE ALLOTTED AMOUNT OF TIME THE NEXT LINE OF THE USING FILE IS SKIPPED. IF THE EXPECTED RESPONSE IS NOT RECEIVED THE LINE IMMEDIATELY FOLLOWING THE WAIT LINE IS PROCESSED. This provides a means for executing a conditional 'GOTO' command depending on whether an expected response is received. Backward branching can be accomplished using the BACKTO command.





When STRIP is OFF, all eight bits of a received byte are used in commands such as the WAIT command. This can lead to situations where a character with the high bit set DOES NOT MATCH a character with the high bit clear, but on the screen both characters look the same!

When VersiCOMM is executing a script file, a  can have differing results depending on whether or not a WAIT command is being executed.

When a  is keyed during a WAIT, but before the WAIT clears, VersiCOMM aborts the WAIT and executes the next line of the script file, just as if the WAIT had timed out. It also sets a flag that indicates a  was keyed. Any following CTRLC command will detect that a  is pending.

When a  is keyed and a WAIT is not being processed, VersiCOMM aborts the execution of the script file and returns to Conversational mode. Any following CTRLC command is not effective, because the script has been aborted.

NOTE: The CCENDTLK OFF command forces COMM to continue to execute the script file when a  is keyed, and a WAIT is NOT being processed. A flag is also set so that any following CTRLC command detects that a  has been keyed. The default for CCENDTLK is ON.

**>WAITIME number**

**>WAITIME 20 (20 seconds)**

Resets the length of time VersiCOMM will WAIT for a character in a USING file before timing-out and returning to the AMOS monitor. Initially this value is about 40 seconds, and this command will reset the value to about 'number' seconds. This command would normally only appear in a USING file.

The WAITIME value is preserved when nested TLK files are executed.

**>WASIT wait-string**

**USING FILES ONLY**

The \$WASIT command works just like the \$WAIT command, except that instead of waiting for new characters from the remote source it "replays" the first 512 characters received while the last \$WAIT command was active.

In effect this allows a USING file to act on any of several responses that could be received from a host.



**>WPAUSE {ON or OFF}**

If turned ON this switch enables "write pausing" in the WRITE and WRITECR commands. When ON, these commands will pause approximately 0.5 seconds when a "write pause character" is encountered. The default "write pause character" is a "~", ASCII 126.

**>WPAUSECHR ascii-code or character**                      **>WPAUSECHR 64 or >WPAUSECHR @**

Redefines the "write pause character."

You may specify either the ASCII code for the character, the character itself, or a VersiCOMM flag that contains the ASCII code for the character. Control characters are indicated by a preceding '^'. VersiCOMM flags are indicated by the character string "FLGn," where n can be 0 to 15.

**>WPREFIX {ON or OFF}**

If turned ON this switch enables "write prefixing" in the WRITE and WRITECR commands. When ON, these commands will convert the next character in the sequence written to a control character when a "write prefix character" is encountered. The default "write prefix character" is a "\*", ASCII 42.

**>WPREFIXCHR ascii-code or character**                      **>WPREFIXCHR 94 or >WPREFIXCHR ^**

Redefines the "write prefix character."

You may specify either the ASCII code for the character, the character itself, or a VersiCOMM flag that contains the ASCII code for the character. Control characters are indicated by a preceding '^'. VersiCOMM flags are indicated by the character string "FLGn," where n can be 0 to 15.

**>WRAP {ON or OFF}**

WRAP is used with the VT100 emulation to set Autowrap mode ON or OFF. If ON, characters that follow the 80th character on a line are wrapped automatically to the next line, otherwise they overwrite the last character on the line.

**>WRITE string**    **>WRITE @0**

VersiCOMM has the capability to pass up to three 30 character strings and 13 60 character strings from the VersiCOMM command line to commands in a USING file. For example,

```
.COMM USING TALK WITH FILE1.LST FILE2.LST FILE3.LST 555-1212
```

would make available the strings FILE1.LST, FILE2.LST, FILE3.LST and 555-1212 to a VersiCOMM USING file. You would specify these strings using the "tokens" @0, @1, @2, and @3 in their place in the USING file. For example,

```
$LOOKUP @0  
$GOTO 100  
$TRANS @0
```

The @3 token is reserved for use by the DIAL command. Tokens greater than @9 are referred to as @A, @B, @C, @D, @E and @F.

These tokens can only be used as arguments to USING file commands, i.e. you could not put '@0' on a line by itself and expect it to get written to the modem port. The WRITE command overcomes this limitation. The actual string, or string token, specified as an argument to this command will be written to the modem port.

WRITE can also be used to send characters, such as '\$' and ';' that have special meaning when they occur in an USING file line.

If the WPREFIX switch is ON, WRITE will convert the next character in the sequence written to a control character when a "write prefix character" is encountered. The default "write prefix character" is a "\*", ASCII 42. So to output a Control-C you could use the sequence "\*C". Note that WRITE can also output control characters entered directly in a WRITE argument line.

If the WPAUSE switch is ON, WRITE will pause approximately 0.5 seconds for each "write pause character" encountered. The default "write pause character" is "~", ASCII 126.

The WRITE command DOES NOT send a terminating carriage-return. Use the WRITECR command if a carriage-return should be sent.

#### **>WRITECR string**

#### **>WRITECR @0**

This command is identical to the WRITE command except that WRITECR command DOES send a terminating carriage-return.

#### **>X command**

#### **>X TYPE \*,\***

The X COMMAND gives non-wildcard file commands full wildcarding capability. It works with COMPIL, TXTFMT, TYPE, VUE, DUMP, SIZE and most other directly executable file-oriented command on your system. It can be executed at VersiCOMM or AMOS command level.

#### **>XORFLG n m**

#### **>XORFLG 0 5**

The XORFLG command is used to logical "xor" a value m to the value contained in the VersiCOMM flag n. The value, m, may range in value from 0 to 255. The flag number n may be 0 to 15.

Both n and m may also be VersiCOMM flags. For example if flag #1 contains 0 and flag #2 contains 5, then XORFLG FLG1 FLG2 is the same as XORFLG 0 5.

These flags are included in the information that can be exchanged with AlphaBASIC using the VersiCOMM-PLUS interface.

**>ZAUTO {ON or OFF}**

The ZAUTO command is used to turn automatic ZMODEM downloading ON or OFF. The default is ON.

If ZAUTO is ON (the default), VersiCOMM will begin to download the file automatically after the host computer starts the ZMODEM transfer.

If ZAUTO is OFF, you will need to switch to VersiCOMM command mode and enter the command RECZ followed by a . See the RECZ command reference for more information.

**>ZDLE {ON or OFF}**

ZDLE is used with the terminal emulations to determine if the ZMODEM link escape character ZDLE (ASCII 30) is passed through the emulator. If the emulator traps ZDLE then the ZMODEM receive program RECZ cannot autostart when the remote computer initiates a ZMODEM transfer.

If ZDLE is ON, then the ZDLE control character is passed through. The default setting is OFF.

Note that ZAUTO must also be ON for RECZ to autostart. This is the default.



# VersiCOMM on the Internet

---

In recent months there has been a great deal of excitement and publicity regarding the Internet. The Information Superhighway is finding its way into a growing number of businesses and homes. Online services such as America Online, Compuserve and Delphi are opening access to the Internet to millions and millions of subscribers. Software packages such as "Internet in a Box" and Netcom's "NetCruiser" have added many more.

Much of the excitement has been generated by graphical applications and resources on the Internet, and these are truly impression. But lost in the clamor of the art shows and weather maps are very useful non-graphical applications and information that for many years have formed the core of the Internet. Even very graphic applications such as World Wide Web (WWW) browsers, have a split personality and can be very useful in a non-graphical environment.

VersiCOMM includes scripts and other support to access the Internet via Netcom Online Communication Services from a Unix shell account. Using a shell account you can send and receive email, read and post to newsgroups, browse the WWW, transfer files, and more, all from the comfort of your Alpha Micro.

While not as glitzy as Windows based products, the capabilities of the Unix Internet applications often exceed their Windows counterparts.

## Scope of This Chapter

The scope, and intent, of this chapter is to introduce the Unix tools used to access information on the Internet. It is not intended to be a treatise on Unix tools and Internet applications.

## What Terminal Should I Use?

It is recommended that a terminal such as an AM65, with "non-space" reverse, underline, and blinking attributes, be used for Internet access. While field terminals such as an AM62 can be used, the additional attributes make some applications more attractive and easier to

use. Often an application will have options for dealing with terminals with a limited number, or no attributes. An example of this is the A)rrrow option in `elm`.

## Establishing an Account

Before you can access the Internet using VersiCOMM you must establish an account with an Internet provider. VersiCOMM includes access scripts and emulation support to access the Internet via a Netcom shell account. VersiCOMM has been used extensively with many of Netcom's shell account applications. Contact Netcom at 800-353-6600 for information about establishing a shell account.

Access via other Internet providers has not been tested, but should be similar to using Netcom.

## Setting Up the Script

After you have established an account you will need to set up the NETCOM script file to use your access number, user name and password. There are two ways you can do this, manually or using the COMMCENTRE..


### Creating the Files Manually

If you want to create the necessary files manually, log to DSK0:[7,121] and create the following files, with the contents indicated.

NETCOM.PHN	The Netcom phone number.
NETCOM.BPS	The port baud rate.
NETCOM.MNP	Create this file to use error correction, otherwise erase.
NETC1.PSW	Netcom user name.
NETC2.PSW	Netcom password.

If your modem is set to use negotiate error correction and compression options as a default, it is not necessary to create the NETCOM.MNP file. In either case the modem control files must have been generated using the DIALER, as described in the VersiCOMM Installation Manual.

## Using the COMMCENTRE

At the AMOS prompt, enter the command "CENTRE" followed by a .

When the COMMCENTRE menu is displayed, select number 65, Setup Service Access. A nearly identical menu, the Access Parameter Definition menu, will be displayed. From this menu select 41, Netcom.

When prompted enter the local Netcom access phone number, the port baud rate, whether or not to use error correction, your username, and finally your password. If your modem uses error correction by default, it doesn't matter whether you answer "yes" or "no" to the "MNP?" prompt.

When this information has been entered you can return to the main COMMCENTRE menu by selecting 66, Return to COMMCENTRE.

## Accessing Netcom

After the script has been set up, Netcom can be accessed several ways:

- Selecting 41, Netcom, from the COMMCENTRE menu.
- Using the command "NETCOM" at the AMOS prompt, or from another menu or program.
- Using the command "COMM modem USING NETCOM" at the AMOS prompt, or from another menu or program.

## The Netcom System

As of May 1995, Netcom was running Sun Release 4.1.3. To obtain current release information you can use the `uname` command:

```
uname -a
SunOS netcom18 4.1.3_U1 9 sun4m
```

## The login Shell

The default login shell (command processor) is the C shell, `csh`. The C shell executes the commands in the file `.login` when you log in and `.logout` when you log out.

An overview of the use of the C shell can be displayed by entering the command:

```
man csh
```

Several alternative shells are also available.

## Logging Off

To logoff Netcom, return to the Netcom prompt and type "logout" and then exit from VersiCOMM by either:

- Entering the Direct Exit character. The default Direct Exit character is the tilde, "~".
- Switching to Command mode using the Change Mode character and entering the command "QUIT." The default Change Mode character, <CTRL><^>, is generated by the <HOME> key on many terminals.

## Getting Help

There are two commands that can be used to get "how to" information, `man` and `faq`. In addition, selected applications have detailed user manuals that can be viewed or downloaded.

### man

The `man` command has two common formats. If you know the name of the command that you need help on, just type `man` followed by the command name:

```
man uname
```

If you don't know the command, use a keyword as in the following example:

```
man -k ftp
```

In this case the query on the keyword "ftp" returns:

```
archie (1)          - query the Archie anonymous FTP databases...
archie (1)          - query the Archie anonymous FTP databases...
ftp (1)             - ARPANET file transfer program
ftp (1C)            - file transfer program
ftpd (8)            - DARPA Internet File Transfer Protocol server
ftpd, in.ftpd (8C)  - TCP/IP Internet File Transfer Protocol server
ftpusers (5)        - list of users prohibited by FTP
mftp (1)            - recieve FTP data via mail
netrc (5)           - file for ftp remote login data
tftp (1C)           - trivial file transfer program
tftpd, in.tftpd (8C) - TCP/IP Trivial File Transfer Protocol server
```

The `apropos` command is the same as the `man` command with `-k`. For example,

```
apropos ftp
```

is equivalent to:

```
man -k ftp
```

For additional information concerning the `man` command, enter the command `man man` at the Netcom prompt.

To obtain a list of common commands, use the command `intro`.

## faq

The `faq` command can be used to obtain information regarding "frequently asked questions" and other topics of general interest.

The command `faq index` will generate a list of `faq` keywords. Following is an excerpt from the index listing:

abuse	account	accounting	accounts	active
adaptor	address	admin	agreement	america
anonymous	ansi	answer	aol	applelink
apropos	arch	archie	archive	archives
arrow	arrows	article	articles	asked
attachment	backspace	backup	bash	biff
big	billing	bills	bin	binaries

The command `faq subject` will generate a list of `faqs` listed by number and subject. For example:

```
0:How to use the FAQ
1:The 'rz' command leaves ^M and ^Z in my text file
2:What are the local dial-in phone numbers for NETCOM?
3:What is NETCOM's office number?
5:How do I remove a file that starts with a minus?
6:Who do I contact about: billing? news? email? uucp? support?
7:Is there any on-line info about NETCOM and its services?
8:What is e-mail and how do I use it?
9:What is Usenet and what do I do with it?
```



## Other Sources of Information

### Unix System

Additional information can be found in the directories under `/usr/local/doc/`. For example, the following files will be found in `/usr/local/doc/usenet/`:

```
A_Guide_to_Social_Newsgroups_and_Mailing_Lists
A_Primer_on_How_to_Work_With_the_Usenet_Community
Answers_to_Frequently_Asked_Questions_about_Usenet
Emily_Postnews_Answers_Your_Questions_on_Netiquette
Hints_on_writing_style_for_Usenet
How_to_Create_a_New_Usenet_Newsgroup
How_to_Get_Information_about_Networks
Rules_for_posting_to_Usenet
What_is_Usenet?
```

The application `utree` can be used to easily view or download these files.

### World Wide Web

As you learn to use the tools described in this chapter, the best source of information becomes the Internet itself. Particularly useful in this regard is *John December's list of Internet Resources* WWW page:

```
Orientation List Internet Orientation
    What is the Internet? * Surfing the Internet * Netiquette *
    Xerox Overview * CyberAtlas * (See also)
*
Guides List Guides to Using the Internet
*   EFF's Internet Guide * EARN's Guide to Network Resource Tools *
    Entering the World-Wide Web * Zen and the Art of the Internet *
*   The Online World * Email 101 * UNIXhelp * (See also)

Reference List Internet Reference
    Information Sources * Internet Tools * Special Internet
    Connections * InterNIC * Merit * Multimedia Index * Internet
    Services FAQ

Explore List Internet Browsing and Exploring
    Gopher * Hytelnet * Global Network Navigator * World Window *
    Internet Hunt * Cultural Aspects * World Wide Web Exhibits

Subjects List Subject-Oriented Searching
    WWW Virtual Library * EINet Galaxy * Yahoo * Subject-Oriented
    Internet Guides * Gopher Jewels * O'Reilly's Whole Internet
    Catalog * USENET Frequently Asked Questions Archive * (See
    also)
*
Words List Word-Oriented Searching
*   Web Search (Catalog) / CUSI / External info * WWW Wanderers
    (Lycos, Harvest, Crawler) * Gopher Jewels Search * WAIS
*   Directory of Servers * Knowbot * World-Wide Web Servers *
    ArchiePlexForm * (See also: keywords, people)

People List Connecting with People
    Academic Discussion Lists * General Discussion Lists *
    Directory Services * Usenet Newsgroups * Internet Relay Chat *
    Multiple User Dialogues * People Lists (directories, home
    pages) * (See also: CMC Forums)
```

This can be accessed using `lynx`.

## Gopher

The following items that can be accessed via the "Internet information" directory of the Netcom gopher:

```
Internet information


--> 1.  Internet Assistance - Collected Resources (peg.cwis.uci.edu)/
    2.  Computer Networks and Internet Guides (Rice U)/
    3.  Clearinghouse of Subject-Oriented Internet Resource Guides (una.hh../
```


Each of these directories contains a substantial list of additional reference material.

## Usenet

The newsgroup **news.newusers.questions** has many questions, and some answers, to a wide variety of Internet topics.

## Command Recall

The command recall system in the C Shell is referred to as "history substitution". It is somewhat tedious compared to the simple  approach of AMOS. To display a list of the last ten commands you have entered in reverse order use the command:

```
history -r 10 
```

To recall and execute the command *n* from the history list enter,


```
!n 
```

where *n* is the number of the command specified in the history listing.

To recall and execute the last command enter:

```
!! 
```

To recall the last command, substitute *str1* for *str2* and execute it, use:


```
!!:s/str1/str2 
```

Individual words from command lines can also be recalled. Refer to the manual pages for `csh` for more information.

The T Shell, `tcsh`, supports command line editing using the cursor keys to scroll the history list and edit commands. Refer to the manual pages for `tcsh` for more information.

## Command Aliasing

Using command aliasing you can substitute a new command name for a command line. For example, the default `.login` file includes the following command:

```
alias lo logout 
```

This defines `lo` to be another name for the `logout` command. An entire command line can be specified in the alias definition. For example:

```
alias h history - r 10
```

## Editing a File

Trying to edit a file on a Unix system can be a shocking experience for an Alpha Micro user, especially if the editor `vi` is used. The `pico` editor is more like VUE. By default, the file manager `utree` uses `pico` for viewing and editing files, and the e-mail application `pine` also uses the `pico` editor.

## Introducing pico

Like VUE, `pico` is a full screen editor that responds to the cursor keys for cursor positioning and uses control sequences to initiate various functions. Control keys for common functions are listed along the bottom of the screen. Most of the functions work as you would expect, with the possible exception of Cut and Uncut.

The best way to get used to the editor is to use it. To start `pico` just enter the command:

```
pico
```

To edit a specific file enter `pico` followed by the filename:



















```
pico station.inv
```

A sample `pico` screen is shown below:





UH PICO(tm) 2.3				File: station.inv				Modified			
01	5749	AL	Muscle Shoals FRA AP	12	1940	MD	34	45	85	37	
01	8024	AL	Talladega	09	1893	PM	33	26	86	05	
02	0080	AZ	Ajo	01	1915	AM	32	22	112	52	
02	3180	AZ	Fort Valley	01	1909	AM	35	16	111	44	
02	5487	AZ	Mesa Exp Farm	03	1896	AM	33	25	111	52	
02	5652	AZ	Yuma Citrus	09	1920	AM	32	37	114	39	
03	1596	AR	Conway	01	1897	AM	35	06	92	25	
03	5820	AR	Pocahontas I	04	1894	PM	36	16	90	58	
04	6074	CA	Napa St Hospital	01	1893	PM	38	17	122	16	
04	6399	CA	Ojai	01	1906	PM	34	27	119	14	
04	9122	CA	Ukiah	01	1893	PM	39	09	123	12	
05	3662	CO	Gunnison	07	1893	PM	38	32	106	56	
08	0211	FL	Apalachicola WSO AP	10	1903	PM/MD	29	44	84	59	
08	0611	FL	Belle Glade Exp Stn	05	1924	AM	26	39	80	38	
08	7851	FL	Saint Leo	03	1895	PM	28	20	82	16	
09	7847	GA	Savannah WSO AP	01	1874	MD	32	08	81	12	
10	1380	ID	Caldwell	10	1904	PM	43	40	116	41	
10	7386	ID	Priest River Exp Stn	02	1898	PM	48	21	116	50	
11	0072	IL	Aledo SNNE	01	1901	PM	41	14	90	44	

`pico` screen for file `station.inv`











The following functions are supported:

	Display this help text.
	move Forward a character.
	move Backward a character.
	move to the Previous line.
	move to the Next line.
	move to the beginning of the current line.
	move to the End of the current line.
	move forward a page of text.
	move backward a page of text.
	Search for (where is) text, neglecting case.
	Delete the character at the cursor position.
	Insert a tab at the current cursor position.
	To invoke the spelling checker
	Report current cursor position
	Insert an external file at the current cursor position.
	Output the current buffer to a file, saving it.
	Exit <code>pico</code> , with option to save buffer.
	Uncut (paste) last cut text inserting it at the current cursor position.

The following functions are supported but cannot be initiated using the standard `pico` control sequences when accessing Netcom using the VT100 emulation.

	Mark cursor position as beginning of selected text. Note: Setting mark when already set unselects text.
	Cut selected text (displayed in inverse characters). Note: The selected text's boundary on the cursor side ends at the left edge of the cursor. So, with selected text to the left of the cursor, the character under the cursor is not selected.
	Format (justify) the current paragraph. Note: paragraphs delimited by blank lines or indentation.
	Refresh the display.

Later versions of the VT100 keyboard translation files (AM62A.VT1, etc.) use the following function keys to generate the required sequences:

	generates		Mark cursor position.
	generates		Format (justify) the current paragraph.
	generates		Uncut (paste) last cut text.
	generates		Cut selected text.
	generates		Refresh the display.

If the keyboard translation file you are using does not include these definitions, the VersiCOMM utility VPSTRN can be used to associate the desired function keys with the appropriate control sequences. See the **Terminal Emulation** chapter of this manual for more information regarding the use of VPSTRN.

## Additional Information

Additional information regarding `pico` can be obtained by entering the commands:

```
man pico
```

and

```
faq pico
```

A short demonstration of `pico` is included in the Internet demo file `INETDM.LIT` in the VersiCOMM Updates section of AMTEC+.

## File Management

### ls

Unless you're familiar with Unix, the first command you probably tried to use in your new account was `DIR`. You could always depend on the `DIR` command to work, whatever the system, whatever the environment. But not in Unix. In Unix you use `ls` to list the contents of a directory.

Many options are defined for `ls`. For a complete listing refer to the manual pages, but here are few to get you started:

- a List all entries; including entries that begin with a ``.``.
- l List in long format.
- R Recursively list subdirectories.
- s Give size of each file.
- t time modified.
- u Sort by time of last access.
- p Put a slash (`/`) after each directory.

You can define `dir` as an alias for `ls` with the `-a` and `-p` options by using the command:

```
alias dir "ls -a -p"
```

## Additional Information


Additional information regarding `ls` can be obtained by entering the command:

```
man ls
```

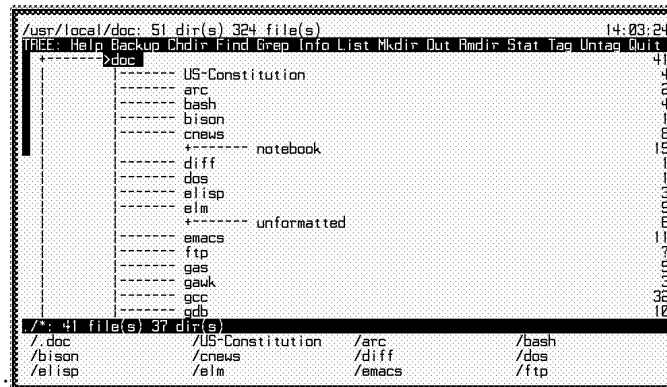
### utree

`utree` is a very useful "screen oriented filesystem browser and utility" similar in application to `OS/Exec` on the Alpha Micro. Using `utree` you can move from directory to directory and



from file to file. Files can be viewed, edited and manipulated in various ways. For example, to examine the directories and files in `/usr/local/doc/` you would enter the command:


```
utree /usr/local/doc/ 
```

A screen similar to the following will be displayed:
















```
utree screen for /usr/local/doc/
```

To exit utree key  or use  as an escape key to exit from the current screen and eventually the application.

To move to a particular directory use the cursor keys. The first few items contained in the directory are displayed in the bottom section of the window. To change to the file screen for the highlighted directory press  or the spacebar.

Following is a summary of some of the more common file screen commands. In general, the uppercase letter applies the operation to all tagged files, while the lowercase letter only applies the operation to the currently highlighted file.

	Tag	Add the highlighted file to the set of tagged files.
	Tag pattern	Add the files matching a specified pattern to the set of tagged files.
	Untag	Remove the highlighted file (lowercase) or all tagged files (uppercase) from the set of tagged files.
		
	Zoom	Display a subset of files matching a specified pattern.
	View	View the highlighted file (lowercase) or all tagged files (uppercase).
		
	Edit	Edit the highlighted file (lowercase) or all tagged files (uppercase).
		
	Status	Show status information for the highlighted file (lowercase) or all tagged files (uppercase).
		
	Copy	Copy the highlighted file (lowercase) or all tagged files (uppercase).
		

	Move	Move the highlighted file (lowercase) or all tagged files (uppercase).
	Remove	Erase the highlighted file (lowercase) or all tagged files (uppercase).
	eXecute	Apply a specified command to the highlighted file (lowercase) or all tagged files (uppercase).

## Downloading Files Using utree

One or more files can easily be downloaded by using the X command. After tagging the files you wish to download, press and enter the command "sz" when prompted. If the VersiCOMM-Plus parameters ZAUTO and ZDLE are ON, VersiCOMM will automatically begin downloading the files, otherwise it will be necessary to switch to Command mode and enter the command RECZ followed by a .

Utree has an extensive Help system that can be accessed by pressing .

## Additional Information

Additional information regarding utree can be obtained by entering the command:

```
man utree 
```

A short demonstration of utree is included in the Internet demo file INETDM.LIT in the VersiCOMM Updates section of AMTEC+.

## Browsing the World Wide Web

The World Wide Web, or WWW, is a virtual Internet environment comprised of pages containing text, graphics, and (usually) links to other pages. WWW pages are constructed using HyperText Markup Language (HTML). HTML has various elements with which you will become more familiar as you travel the Web.

The WWW traveler uses the Unix program "lynx" to display the contents of these pages and to move deftly from page to page by selecting highlighted text links. In addition to communicating via HTTP to Web servers, lynx can also provide information from Gopher, FTP, NNTP, and WAIS servers.

The WWW is an amazing place, brimming with information, that can benefit just about any user with any interest. Let's look at a couple of examples.

What was the temperature in the Sierra Nevada today at an elevation of 11,500 feet at Upper Tyndall Creek? The answer is just a couple of links away on the *California Department of Water Resources Snow Page*:

06:00:00	27.6 deg f
07:00:00	30.6 deg f
08:00:00	34.3 deg f
09:00:00	36.9 deg f
10:00:00	39.2 deg f
11:00:00	40.6 deg f
12:00:00	40.5 deg f
13:00:00	44.4 deg f
14:00:00	43.6 deg f
15:00:00	45.8 deg f

What days are holidays in Japan? A link to the *Prime Minister's official residence* will tell us:

January 1	(Ganjitsu) / New Year's Day
January 15	(Seijin-no hi) / Adult's Day
February 11	(Kenkoku Kinen-no hi) / National Founding Day
March 20 or 21	(Shunbun-no hi) / Vernal Equinox Day
April 29	(Midori-no hi) / Greenery Day
May 3	(Kenpou Kinenbi) / Constitution Memorial Day
May 4	(Kokumin-no Kyuujitsu) / National People's Day
May 5	(Kodomo-no hi) / Children's Day
September 15	(Keirou-no hi) / Respect-for-the-Aged Day
Sept. 23 or 24	(Shuubun-no hi) / Autumn Equinox Day
October 10	(Taiiku-no hi) / Sports Day
November 3	(Bunka-no hi) / Culture Day
November 23	(Kinrou Kansha-no hi) / Labor Thanksgiving Day
December 23	(Tennou Tanjoubi) / Emperor's Birthday

Do you have a dog? Web pages exist for many breeds. Just in case you're interested, here's some information about the 1995 National dog show for Golden Retrievers:

Show Headquarters is the Red Lion Inn at Rohnert Park, California.  
This is a convention type hotel with spa, pool, tennis courts, etc.  
There are 200 rooms available at show rates. The dates are:

Field Trial	Sept 12-14	Martis Creek Lake, Truckee, CA
WC/WCX	Sept 16	Grizzley Island near Vallejo
Tracking	Sept 17	Hastings Island near Rio Vista
Sweepstakes, Conformation		
Obedience	Sept 18,19	Rohnert Park Red Lion
Conformation	Sept 20	


Excuse me for a second...FedEx is at the door... Do you ever ship anything via Federal Express? Do you have packages in transit at the moment? Want to know where they are? Just follow the instructions on the *FedEx Airbill Tracking Form Page*:


You can track the status of your package anytime of day, anywhere in the world. You'll be able to follow your package's journey even while it is still in transit. Type in your FedEx package tracking number in the field below and the delivery and/or the scan information for the package will be displayed. When entering a tracking number, remember to include any dashes that appear in the number (international numbers begin with 400- and COD shipments begin with 300- or 320-). Select "Send Request".

The information on the WWW is incredibly diverse. It ranges from the seriously scientific to the absolutely absurd. But as the old Japanese proverb says, "Juu-nin to-iro" or "Ten people, ten colors."

## Starting lynx

From the Netcom prompt enter the following command:

```
lynx 
```

After the default page is displayed, press the  key and when prompted for a "URL" enter the following address:

```
http://www.netcom.com/netcom/assist.html
```



You will not have to do this each time you browse the Web, but for now carefully enter the address and press return.

This will transfer you to the *NETCOM -- Internet Assistance and Information* page. Your screen should appear similar to:

```

NETCOM -- Internet Assistance and Information

INTERNET ASSISTANCE AND INFORMATION
* John Decker's List of Internet Resources
* An Index to Web Pages
* The InterNIC (Network Information Center)
* The Virtual Reference Desk from UC Irvine
* UMich SILS Subject Reference Page
* Scott Yanoff's List of Internet Resources
* World Wide Web Starting Points Document
* NCSA Mosaic Demo Document
* The World Wide Web Overview
* The What's New Page
* Arlene Rinaldi's Netiquette paper
* Help with UNIX Commands
* Help with mailing list programs (Listservers)
* Help from the Internetwork Mail Guide

Return to NETCOM Home Page
webspin@netcom.com 2 December 1994

Commands: Use arrow keys to move, ? for help, q to quit, <- to go back
Arrow keys: Up and Down to move, Right to follow a link, Left to go back.
H)elp Q)ptions P)rint G)o M)ain screen Q)uit /-search [delete]-history list

```

### NETCOM -- Internet Assistance and Information Page

When this page is displayed press **A**. The following message will be displayed near the bottom of the page:

```
Do you wish to save this document in your current bookmark file? (y/n)
```

Press the **Y** key. Lynx will respond with "Done!" when the bookmark has been added.

You can always return to this page by pressing the **V** key, then selecting *NETCOM -- Internet Assistance and Information* from the bookmark list and pressing **J**.

The **Defining Your Home Page** section below describes how to make a particular page the first one that lynx displays.

The *NETCOM -- Internet Assistance and Information* page is a good place to start exploring the Internet.

### I'm Waiting...and Other Hints

Sometimes a jump to a WWW page will fail, or will be incredibly slow due to various factors. If you don't want to wait for the connection attempt to timeout, press **Ctrl G** or press **Z** to abort the attempt.

On occasion a WWW screen will be garbled or include extraneous characters. Press **Ctrl W** to refresh the screen or press **Ctrl R** to reload and refresh the screen.

To get Help press **?** or **H**. Note that the help information is not stored in lynx but is retrieved from a WWW site at the University of Kansas.

To exit lynx press **Q**. An uppercase **Q** does a "quick Quit."

## lynx Commands

The following commands are used for navigating from one page to another and within the currently displayed page:

	Highlight next link topic
	Highlight previous link topic
or	Jump to WWW page for highlighted topic
	Return to previous WWW page
or space	Scroll down
or	Scroll up
	Add the current link to your bookmark file
	View your bookmark file
	Goto a user specified URL or file.
	Return to main screen
backspace	Go to the history page

These additional commands are also defined:

	Send a comment to the document owner
	Download the current link
	Edit the current file
	Show an index of documents
	Set your options
	Print to a file, mail, printers, or other
	Search for a string within the current document
	Enter a search string for an external search.
	Go to the next search string
	Show file and link info
	Toggle document source/rendered view
	Spawn your default shell
	Erase input line

## Defining Your Home Page

To define a particular WWW page as your "home" page use the command:

```
setenv WWW_HOME URL-address
```

*URL-address* is the location of the page. For example:

```
setenv WWW_HOME http://www.netcom.com
```

This command can be placed in your `.login` file so it is executed automatically when you log in.

## Yahoo and Other Search Utilities

If you did some exploring starting at the *NETCOM -- Internet Assistance and Information* page, you may have noticed a reference to "Yahoo." Yahoo is a WWW search tool that can be accessed from *John December's list of Internet Resources*. It is found in the following group of links:

Subjects List Subject-Oriented Searching  
 WWW Virtual Library \* EINet Galaxy \* **Yahoo** \* Subject-Oriented

Select the Yahoo link and a page similar to the following is displayed:

```

                                YAHOO

[ What's New? | What's Cool? | What's Popular? | A Random Link ]

[ Yahoo | Up | Search | Suggest | Add | Help ]
* Art (934) [new]
** Business (11989) [new]
* Computers (3828) [new]
* Economy (1169) [new]
* Education (2172) [new]
* Entertainment (12508) [new]
* Environment and Nature (331) [new]
* Events (86) [new]
* Government (1387) [new]
* Health (821) [new]
* Humanities (344) [new]
* Law (283) [new]
* News (414) [new]
* Politics (275) [new]
-- press space for next page --
Arrow keys: Up and Down to move. Right to follow a link; Left to go back.
H)elp O)ptions P)rint G)o M)ain screen Q)uit /=search [delete]=history list

```

Before doing a Yahoo search add the Yahoo page to your bookmark list by pressing **A**, and then **Y** in response to the add prompt.

Next use the **↓** key to move to the **| Search |** link and select it by pressing **↓**. The Yahoo Search form should display. This form illustrates some features of HTML you may not have used before, such as a text entry field, form buttons, and checkbox fields.

Use the **↓** key to move to the text entry field. In this field type the word "whitehouse" (no quotes) and then use the **↓** key to move to the Search button adjacent to the text entry field and then press **↓**. After a few moments (usually) Yahoo will return a list of WWW pages that contain the search term or terms. Did Yahoo find anything interesting?

Note that Yahoo also has subject category links and links to other search tools. These can be very useful when trying to locate WWW pages of interest.

## Additional Information

Additional information regarding lynx can be obtained by entering the commands:

```
man lynx ↓
```

and

```
faq lynx ↓
```

Much additional information regarding the WWW can be found on the Web. A good starting point is *John December's list of Internet Resources* WWW page. Another good source of information are the documents that can be accessed from the "Internet information" directory of the Netcom gopher.

A demonstration of lynx is included in the Internet demo file INETDM.LIT in the VersiCOMM Updates section of AMTEC+.

## Browsing the Net Using gopher

The Netcom gopher server describes `gopher` as "software following a simple protocol for tunneling through a TCP/IP internet, and running errands, especially the retrieval of documents."

Gopher is basically a system of menus. Menu entries can be links to other menus or to data, usually in the form of text.

### Terminal Notes

Menus containing lines longer than 80 characters may not display correctly unless the VersiCOMM WRAP parameter is OFF and your terminal's AUTOWRAP feature is turned OFF.

To turn VersiCOMM's emulation WRAP OFF, switch to Command mode and at the prompt enter the command:

```
WRAP OFF
```

### You Have to Start Someplace

At the Netcom prompt enter:

```
gopher
```

A menu similar to the following will be displayed:

```

Internet Gopher Information Client 2.0.pl5
Root gopher server: gopher.netcom.com

--> 1. About this GOPHER server.
    2. Information about NETCOM/
    3. Internet information/
    4. Jughead - Search High-Level Gopher Menus (via Washington & Lee) <?>
    5. Search Gopherspace using Veronica/
    6. Other Internet Gopher Servers (via U.C. Santa Cruz)/
    7. Weather (via U. Minnesota)/
    8. Worldwide Directory Services (via Notre Dame)/
    9. Interesting items/
   10. ATTENTION NETCOM users.

Press ? for Help. q to Quit. u to go up a menu
Page: 1/1


```



### NETCOM -- Root Gopher Server

This is the top menu for Netcom's gopher server.

Using `gopher` is very easy. For example, use the `↓` key to move down to item 10., ATTENTION NETCOM Users, and press `↓` to display information about the "mother gopher" server at the University of Minnesota.














When the last page of a text item has been displayed, the following prompt appears:

Press  to continue,  
 <m> to mail, <D> to download, <s> to save, or <p> to print:






Usually you will just press  and continue. But note there are options to mail, download and save the item you have displayed. For now press  to return to the top Netcom menu.

## Keyboard Controls














### Navigation

-  Move up a line.
-  Move down a line.
-  or  Display current item.
-  or  Go back to previous menu.
  
-  or  or spc View next page.
-  or  or  View previous page.
  
-  Go to a specific line.
-  Go back to the main menu.

### Bookmarks

-  Add current item to the bookmark list.
-   Add current directory/search to bookmark list.
-  View bookmark list.
-  Delete a bookmark/directory entry.

### Other Commands

-  Save current item to a file.
-   Download a file.
-  Quit with prompt.
-   Quit unconditionally.
-  Display Technical information about current item.
-   Change options
-  Search for an item in the menu.
-  Find next search item.
-  Open a new gopher server
-  Shell Escape

## Searching Using Veronica


Veronica is a keyword search tool that is used to locate items that reside on most of the gopher servers on the Internet. From the top menu of the Netcom gopher select **Search Gopherspace using Veronica**. A menu similar to the following will be displayed:

```

Search Gopherspace using Veronica

-->  1.  How to Compose veronica Queries - June 23, 1994.
      2.  Frequently-Asked Questions (FAQ) about veronica - January 13, 1995.
      3.  More veronica: Software, Index-Control Protocol, HTML Pages/
      4.  Simplified veronica: Find Gopher MENUS only <?>
      5.  Simplified veronica: find ALL gopher types <?>
      6.  Find GOPHER DIRECTORIES by Title word(s) (via University of Pis.. <?>
      7.  Find GOPHER DIRECTORIES by Title word(s) (via UNINETT..of Bergen) <?>
      8.  Find GOPHER DIRECTORIES by Title word(s) (via SUNET) <?>
      9.  Find GOPHER DIRECTORIES by Title word(s) (via NYSERNet) <?>
     10.  Find GOPHER DIRECTORIES by Title word(s) (via University of Koe.. <?>
     11.  Find GOPHER DIRECTORIES by Title word(s) (via PSINet) <?>
     12.  Search GopherSpace by Title word(s) (via University of Pisa) <?>
     13.  Search GopherSpace by Title word(s) (via UNINETT/U. of Bergen) <?>
     14.  Search GopherSpace by Title word(s) (via SUNET) <?>
     15.  Search GopherSpace by Title word(s) (via NYSERNet) <?>
     16.  Search GopherSpace by Title word(s) (via University of Koeln) <?>
     17.  Search GopherSpace by Title word(s) (via PSINet) <?>

```


Select any of the **Search GopherSpace by Title word(s)** entries and in the resulting dialog box enter "whitehouse" (no quotes) as the word to search for and press . Don't be surprised if the following "menu" results:

```

Search GopherSpace by Title word(s) (via University of Pisa): whitehouse

-->  1.  *** Too many connections - Try again soon. ***.

```

This is a common response to a Veronica query and occurs when the server is too busy to handle the search request. When this happens, use the  key to return to the **Search Gopherspace** menu and retry the same or a different server. If you're not successful in a few attempts try your search later.

A successful search will produce a menu similar to the following:

```

Search GopherSpace by Title word(s) (via University of Pisa): whitehouse

-->  1.  whitehouse.www.
      2.  United States Government - Agencies, Congress, Whitehouse/
      3.  whitehouse [28Jun94, 4kb].
      4.  whitehouse.gif [28Nov94, 2kb] <Picture>
      5.  president@whitehouse.gov.
      6.  president@whitehouse.gov.
      7.  93-12-14-02: Whitehouse Communications.
      8.  93-12-15-18: Whitehouse e-mail.
      9.  WhiteHouse_Email.txt.
     10.  Whitehouse information.
     11.  whitehouse.press.gopher.
     12.  whitehouse.press.gopher.
     13.  WhiteHouse.html.
     14.  Whitehouse Technology Plan (WORD) <HGX>
     15.  Whitehouse Access.
     16.  whitehouse.
     17.  WWW> New WWW @ Whitehouse announced by Gore (fwd).
     18.  Re. Whitehouse online.

```

In this case this is the first page of a multi-page menu containing over 200 items. You will find that many items are duplicated because they reside on more than one gopher server.

Refer to the *How to Compose veronica Queries* and *Frequently-Asked Questions (FAQ) about veronica* in the *Search Gopherspace* menu for more information.

## A gopher Sampler

### How Is Your Car Running?

Have any recall notices or Technical Service Bulletins (TSB) been issued for the car you drive? You can find out using the ALLDATA automotive repair information gopher.

To access the ALLDATA gopher enter the following command at the Netcom prompt:

```
gopher gopher.alldata.tsb.com
```

A menu similar to the following should be displayed:

```
gopher.alldata.tsb.com
```

1. About the ALLDATA Corporation.
2. See Recall Notice and Technical Service Bulletin (TSB) Titles for ../
3. How TSBs can help you maintain your car.
4. What are Recall Notice Titles, and how can they help you?.
5. Take a Test Drive!/.
6. Free Trial Access to the World's Largest Database of Vehicle Repai...
7. About ALLDATA-LINK for your PC.
8. Download ALLDATA-LINK (for IBM compatibles)/
9. Find Locations of ALLDATA Repair Shops Nearest You (Web client requ..
10. Find Locations of ALLDATA Repair Shops Nearest You (by email).
11. Announcing ALLDATA's WWW Server.
12. Connect to ALLDATA'S WWW Server (Web client required)
13. How to contact us.

Select item 2., *See Recall Notice and Technical Service Bulletin (TSB) Titles*, then just follow the menus, selecting the appropriate year, make and model, and other defining attributes for the car you are researching. Most, but not all models, are listed.


You can also access this information via the WWW using lynx. ALLDATA's WWW server address is:

```
http://www.alldata.tsb.com
```

You will note that ALLDATA also offers a more comprehensive and detailed fee-based dial in service for automotive repair information. The information that is publicly available via the Internet is limited in scope but useful none-the-less.

### Is It Going to Rain Tomorrow?

If you're a weather nut you'll love the information you can glean from gopherspace. There are a number of weather related gophers. As an example, try out the University of Illinois, Department of Atmospheric Sciences, Weather Machine Gopher Server.

If you're already in gopher and at a menu, you can jump to a new gopher by pressing the  key and entering "wx.atmos.uiuc.edu" (no quotes) when prompted for a host name. Alternatively, from the Netcom prompt you can enter the command:

```
gopher wx.atmos.uiuc.edu
```

In either case, you will be presented with a menu similar to the following:

```
wx.atmos.uiuc.edu
```

```
1. :The UofI Weather Machine.
2. Canada/
3. Caribbean/
4. Case Studies/
5. Documents/
6. Illinois/
7. Images/
8. International/
9. Latin America/
10. Other Servers/
11. Regional/
12. Satellite Discussion/
13. Severe/
14. States/
15. Status/
```

Select "States" and on the next menu, select the state in which you live. A listing of available reports will be displayed. For example, here is the "Coast Guard Report" for (southern) California:

ID	WX	VSF	WIND	WAVE	SEA	AIR	REMARKS	STATION NAME
L98	CYF03	/	S03	/	0114	/	55 / 57 / SW	SWL EL CAPITAN BCH
P44	CYF03	/	SW08	/	0110	/	59 / 61 / SW	SWL SANTA BARBARA HBR
L79	CYF03	/	SSW05	/	0212	/	57 / 55 / W	SWL CHANNEL ISLANDS HBR
0L3	CYF02	/	SW04	/	0212	/	57 / 60 / SW	SWL ZUMA BEACH
4L7	CYH04	/	WSW04	/	0212	/	60 / 60 / SW	SWL HERMOSA BCH
L14	CYH04	/	WSW03	/	0209	/	58 / 62 / SW	SWL CABRILLO BCH
6L9	PC08	/	W08	/	0315	/	59 / 58 /	HUNTINGTON BCH
3L3	CYF04	/	SW05	/	0415	/	57 / 56 / S	SWL NEWPORT BEACH
9L0	CY10+	/	SSW04	/		/	59 / 62 /	DANA POINT HBR
L10	CYH06	/	CALM	/	0310	/	60 / 61 / SW	SWL SAN CLEMENTE PIER

## Additional Information

Additional information regarding `gopher` can be obtained by entering the commands:

```
man gopher
```

and

```
faq gopher
```

Not surprisingly, `gopher` can provide you with much additional information about `gopher`. See, for example, the items and documents in the "Internet information" directory of the Netcom `gopher`.

A demonstration of `gopher` is included in the Internet demo file `INETDM.LIT` in the VersiCOMM Updates section of AMTEC+.

## Electronic Mail

Electronic mail is clearly one of the core applications of the Internet. E-mail is generally fast, convenient, and easy to use.



Netcom supports several electronic mail systems, including `elm`, `pine`, `mm`, `mail`, `eudora` and `mush`. The `elm` electronic mail system is introduced here. For more information on the other mail systems use the `man` command. For example:

```
man pine
```

The command,

```
faq mail
```

may also provide some useful information.

## First Things First...

Unless you have another preference, check that `elm` is using `pico` as the default editor. At the Netcom prompt type:

```
elm
```

When prompted for a command press the `o` key to access the `elm` options editor. Check that the following options specify `pico`:

```
E)ditor (primary)      : /usr/local/bin/pico
V)isual Editor (~v)    : /usr/local/bin/pico -t
```

When done, return to the mail index screen by pressing `q` or `Q`, and then press `q` to quit and return to the Netcom prompt.

## Getting Help

A summary of key initiated commands can be obtained by pressing the `?` key when `elm` prompts for a command. Help is also available in some other contexts. Generally `elm` displays the message "Press ? for help" when additional help is available.

## Send a Test Message...

At the Netcom prompt enter the command,

```
elm username
```

where *username* is your Netcom user name. Elm should execute and enter its message composition mode using the default editor. This example assumes the editor is `pico`.

Using the editor, type a message. When it is complete, press `Ctrl X` to exit the editor. The following message will be displayed:

```
Save modified buffer (ANSWERING "No" WILL DESTROY CHANGES) (y/n)?
```

Press `y` to save the buffer and then a `q` to OK the filename to write. The following message is displayed:

```
Please choose one of the following options by parenthesized letter: s
e)dit message, edit h)eaders, s)end it, or f)orget it.
```

Press `q` to send the message. That's all there is to it!

## Read a Test Message...

To read the message you just sent, and any other messages type:

```
elm ␣
```

Elm will display an index of pending messages, one of which should be the test message. If the test message has not been received exit by pressing **Q** and try again in a few minutes.

If more than one message is displayed, used the cursor keys to highlight the test message and then press **␣** to read it. Note that the last message received is the first message listed.

## Reply to the Test Message...

When you have finished reading the message, elm will prompt you for a command. To reply to the message press **R**. Elm will ask if you want to copy the message into the reply. Press **Y**. It will then suggest a subject topic. Press **␣**. Also press **␣** at the "Copies To:" prompt.

Next elm enters edit mode with the previous message written in the buffer. Reply as appropriate and when done press **Ctrl X** to exit the editor. Save the buffer by pressing **Y** and then press **␣** to accept the filename and **␣** to send the reply.

Return to the mail index by pressing **I** when prompted for a command.

## Saving the Test Message After Reading

When you exit elm it will ask you if you want to save the messages you have read in the "received" folder. Press **Y** or **N**, as appropriate.

## Additional Information

A brief overview of elm, including several command line options, can be displayed using the command:

```
man elm ␣
```

More detailed information will be found under the directory `/usr/local/doc/elm/`. Several lengthy guides to using the Elm system will be found here. The filenames are:

Users	Elm Users Guide
Alias	Elm Alias System Users Guide
Filter	Elm Filter System Guide
Form	Elm Forms Mode Guide
Ref	Elm Reference Guide
Config	Elm Configuration Guide (for developers)

For general information regarding email, browse *John December's list of Internet Resources* WWW page and the *Internet information* directory of the Netcom gopher.

## Working with Newsgroups

Newsgroups are like magazines or books. There is a newsgroup for just about any interest or taste. You can be assured there are newsgroups that you would find enlightening as well as those you would find offensive. Newsgroups can serve purposes from the most serious to the most frivolous and everything in between.

The newsreader described here is `rn`. It is a commonly used reader that supersedes `mail` and `mailx`.

The "no news is good news" reader `nn` is also available on the Netcom machine. For an overview of this reader enter the command:

```
man nn
```

## Before You Start (Don't Skip This Step)

Before you start using `rn` execute the following command:

```
setenv RNINIT -q
```

This prevents `rn` from asking whether you should add each of ten zillion new newsgroups to your subscription list with no way to escape.

This command should also be added to your `.login` file.

## Getting Help

When user input is requested `rn` will list common (but not all) responses in square brackets. To review possible responses press `[H]`.

## Your First Newsgroup

The first newsgroup we'll read is **netcom.announce**, a newsgroup for Netcom announcements and other information for subscribers. At the Netcom prompt enter the command:

```
rn netcom.announce
```

The response should be:

```
Newsgroup netcom.announce not in .newsrsrc--subscribe? [ynYN]
```

Press (lowercase) `[y]`. The program will respond with:

```
Put newsgroup where? [^[.L]
```

Press `[.]`. The newsreader will then display a message similar to:

```
*****      4 unread articles in netcom.announce--read now? [ynq]
```

Press `[=]` (equal key). A list of titles for the articles you have not read should be displayed, followed by a prompt asking what to do next:

```
231 "Good Times" Virus
232 Canon selects NETCOM's NetCruiser
233 NN Newsreader Problem Fixed
234 /usr/hack/tmp disk space changes
What next? [npq]
```

Select one of the articles and enter the number at the prompt. In this example we'll enter "231". The reader then displays the first few lines of the article:

```
Article 231 (3 more) in netcom.announce (moderated):
From: nc0022@netcom.com (Margaret)
Subject: "Good Times" Virus
Organization: NETCOM On-line Communication Services (408 261-4700 guest)
X-Newsreader: TIN [version 1.2 PL1]
Date: Mon, 8 May 1995 19:08:12 GMT
Lines: 15

--MORE--(37%)
```

Use the SPACEBAR to view the rest of the article:


```
Recently, several users have expressed concern about receiving notification
of a "Good Times" virus. There is no such virus; these messages are part
of a hoax which has been circulating on the Internet since late last year.

If you would like more information about this urban legend, a FAQ about
it is available in our anonymous FTP directory. To retrieve this file,
ftp to ftp.netcom.com and look in the directory pub/ne/netcom.



Margaret
--

Margaret                                     Netcom On-line Communications
Email Coordinator                           Technical Support
nc0022@netcom.com

End of article 231 (of 234)--what next? [npq]
```

Press the  key to return to the article title listing:

```
232 Canon selects NETCOM's NetCruiser
233 NN Newsreader Problem Fixed
234 /usr/hack/tmp disk space changes
End of article 231 (of 234)--what next? [npq]
```

For the moment, press  to quit reading the newsgroup and then press  again to exit to the Netcom prompt. The "restriction" message that displays just means that `rn` was restricted to reading the specified newsgroup **netcom.announce**.





## Finding Newsgroups to Read

Now that you know how to use `rn` to read a newsgroup, how do you find out what newsgroups to subscribe to?

### Using `rn` to Locate Newsgroups

This is a quick, but messy, way to locate newsgroups. For example, to find all newsgroups that have "netcom" in their title enter the command:

```
rn netcom 
```

For each newsgroup listed, press (lowercase) . There will be quite a few! When you have stepped through a few of the names, instead of pressing lowercase , press uppercase . When the remaining newsgroups have been processed you may get a prompt asking you if you want to read the unread messages in netcom.announce. If so, press . Eventually, the following message is displayed:

```
***** End of newsgroups--what next? [npq]
```

Press **[X]** in response to this prompt. This prevents `rn` from updating the `.newsrc` file and adding unnecessary references. Should it be necessary, you can edit the `.newsrc` file using `pico`.

### Using lynx to Locate Newsgroups

When you learned to use `lynx`, a bookmark was added for the *NETCOM -- Internet Assistance and Information* page of the WWW. Start up `lynx`, and then use the **[V]** key to display your bookmarks. Find *NETCOM -- Internet Assistance and Information* and press **[J]**.

If you have not added this bookmark, press **[G]** and when prompted for the URL enter:

```
http://www.netcom.com/netcom/assist.html [J]
```

When you have successfully linked to the *NETCOM -- Internet Assistance and Information* use the SPACEBAR and **[J]** key to locate *John December's list of Internet Resources*. Select this link and then locate *Usenet Newsgroups* (near the end) and select it. Follow the links according to your interests or curiosity.

Alternatively you can go directly to the *Internet Newsgroups* page by pressing **[G]** and when prompted:

```
http://www.w3.org/hypertext/DataSources/News/Groups/Overview.html [J]
```

### Posting a Reply to a Newsgroup Article

After you've spent some time "lurking" in a newsgroup, or maybe after only a few minutes, you're going to want to post a reply to an article. Maybe you have the answer to a posted question, or you disagree with some advice in a follow up to an article, or you just want to contribute to a discussion. Before posting your first follow-up, consider doing a practice posting in the **netcom.test** newsgroup.

Access the **netcom.test** newsgroup by entering the command:

```
rn netcom.test [J]
```

At the "read now?" prompt press the **[=]** (equals) key. Select an article and read it by entering its number at the prompt Read the article, and at the "What next?" prompt, press **[F]**. The program will respond with an "etiquette" warning and the following message:

```
This program posts news to machines throughout Netcom.
Are you absolutely sure that you want to do this? [ny]
```

Press **[Y]**. The response will be:

```
Prepared file to include [none]:
```

Text from a file may be included by specifying the file name. For now press **[J]**. The program will change to the default editor (usually `pico`) with the text of the reply in the buffer. Add your reply at the end of this text and press **[Ctrl][X]**. Answer **[Y]** to save the buffer, and press **[J]** to accept the filename. The following prompt will be displayed:

Send, abort, edit, or list?

Press **[S]** and your reply will be (eventually) posted. Check later to see if it was posted correctly.

Press **[Q]** at the next two "What next?" prompts to exit `rn`.

## Additional Information

Additional information regarding `rn` can be obtained by entering the commands:

```
man rn
```

and

```
faq rn
```

and

```
faq news
```

The following files will be found in `/usr/local/doc/usenet/` :

```
A_Guide_to_Social_Newsgroups_and_Mailing_Lists
A_Primer_on_How_to_Work_With_the_Usenet_Community
Answers_to_Frequently_Asked_Questions_about_Usenet
Emily_Postnews_Answers_Your_Questions_on_Netiquette
Hints_on_writing_style_for_Usenet
How_to_Create_a_New_Usenet_Newsgroup
How_to_Get_Information_about_Networks
Rules_for_posting_to_Usenet
What_is_Usenet?
```

The application `utree` can be used to easily view or download these files.

The newsgroup **news.newusers.questions** may also have answers to your questions about newsgroups.

A demonstration of `rn` is included in the Internet demo file `INETDM.LIT` in the VersiCOMM Updates section of AMTEC+.

## Terminal Access to Net Sites

Telnet is a program that lets the user work as a terminal on a remote computer using the TELNET protocol. Accessing a site via `telnet` is much like accessing a BBS via a modem using a stripped version of VersiCOMM. Using `telnet` you can work interactively on the remote system and that's about it. Even so, it's a very useful tool.

### Starting a telnet Session

A `telnet` session is initiated by entering the command,

```
telnet address
```

where *address* is the symbolic or numeric Internet address for the site. If required, a port number can be appended to the command line. For example:

```
telnet 111.111.1.11 5000
```

The following will be displayed when telnet connects to *address*:

```
Connected to address.
Escape character is '^]'.
```

Many telnet sites will request a user name and password upon connection.

## Telnet Commands

Entering the escape character, `Ctrl-]` returns the user to the telnet executive and prompt.

Entering a `?` at the prompt, provides the following command summary:

```
telnet> ?
Commands may be abbreviated.  Commands are:

close          close current connection
display        display operating parameters
mode           try to enter line-by-line or character-at-a-time mode
open           connect to a site
quit           exit telnet
send           transmit special characters ('send ?' for more)
set            set operating parameters ('set ?' for more)
status         print status information
toggle         toggle operating parameters ('toggle ?' for more)
z             suspend telnet
?             print help information
```

## An telnet Session to the CDEC

Following is a telnet session to the California Data Exchange Center:

```
telnet 000.000.0.00
Trying...
Connected to 000.000.0.00.
Escape character is '^]'.

**** AOS/VS II  Release 2.21.00.00 / Press NEW-LINE to begin logging on ****

AOS/VS II 2.20.00.00 / EXEC-32 2.21.00.00      21-May-95  8:07:12      @TCON8
Username: xxxxxxxx
Password:
-----
      Copyright (C) Data General Corporation, 1980 - 1993
      All rights reserved.
      Licensed material -- property of Data General Corporation

      This software is made available solely pursuant to the
      terms of a DGC license agreement which governs its use.
-----
      CDEC DATABASE ACCESS SYSTEM  ---  MAIN MENU

SELECT ONE:
  1. Access real-time data for single stations
  2. Access real-time data for a group of stations
  3. Download data in Standard Hydrological Exchange Format (SHEF)
  4.
  5. Access daily and monthly data for single stations (CIMIS also)
  6. Access daily and monthly data for a group of stations
  7. List contents of a file
  8. Enter data into the database (special privileges required)

For help facility, type 'HELP'
```

```

Your request:1
Enter ESC key for default
Starting date(day,hour OR day,hour,month,year):$
Enter ESC key to stop
Station:CHILAO_____

Stream or River Basin: SAN GABRIEL R           Elevation:      5450 ft
                        CHILAO
[CIO]      WIND SP WIND DR RAIN      TEMP      FUEL TP FUEL MS REL HUM
           PDT      MPH      DEG      INCHES    DEG F    DEG F    %    %
05/20/95
    1600      12      199      1.50      67      87      --      99
    1700      10      207      1.50      66      83      --      99
    1800      10      214      1.50      62      73      --      99
    1900       4      229      1.50      58      63      --      99
    2000       2       80      1.50      52      47      --      99
    2100       3       71      1.50      52      46      --      99
    2200       3      125      1.50      51      45      --      99
    2300       2      229      1.50      53      46      --      99
    2400       1      264      1.50      53      48      --      99
05/21  100       2      108      1.50      53      46      --      99
        200       5       92      1.50      49      44      --      99
        300       6       98      1.50      47      42      --      99
        400       3      133      1.50      50      42      --      99
        500       6       96      1.50      47      41      --      99
        600      --      --      --      --      --      --      --
        700      --      --      --      --      --      --      --
        800      --      --      --      --      --      --      --
Enter ESC key to stop
Station:$_____

Type 'MENU' to list option menu
Your request:BY

Process 26 TerminatedSTOP
Elapsed Time   0:00:55, CPU Time   0:00:00.943, I/O Blocks  98

User 'Connection closed by foreign host.

```

A similar telnet session is included in the Internet demo file INETDM.LIT in the VersiCOMM Updates section of AMTEC+.

## Additional Information

Additional information regarding telnet can be obtained by entering the commands:

```
man telnet
```

and

```
faq telnet
```

## File Transfers

File transfers on the Internet are accomplished with `ftp`, a user interface to the ARPANET standard File Transfer Protocol.

## Known, But Anonymous

There are a large number of publicly accessible file archives known as "anonymous FTP sites." They are referred to as "anonymous" because you can log-in as a guest using "anonymous" as a user name. This really is a misnomer, however, because you're expected to supply your full email address as a password. So much for anonymity!



## Starting a ftp Session

A ftp session is initiated by entering the command,

```
ftp address
```

where *address* is the symbolic or numeric Internet address for the site. For example:

```
ftp ftp.netcom.com
```

If the connection is successful you will normally be prompted for a name and password. To log-in to an anonymous ftp site you would use "anonymous" for the user name and your full email address for the password.

## ftp Commands

Entering a `?` at the ftp prompt, provides the following command summary:

```
ftp> ?
Commands may be abbreviated.  Commands are:

!           cr           macdef       proxy        send
$           delete       mdelete     sendport     status
account     debug          mdir        put          struct
append      dir            mget        pwd          sunique
ascii       disconnect   mkdir       quit         tenex
bell        form        mls         quote        trace
binary      get          mode        recv         type
bye         glob        mput        remotehelp   user
case        hash        nmap        rename       verbose
cd          help        ntrans     reset        ?
cdup        lcd         open        rmdir
close       ls          prompt      runique
```

Very terse help may be obtained for an individual command. For example:

```
ftp> ? get
get          receive file
```

Note the command "remotehelp." This command can be useful for determining what ftp commands are available on a remote system.

## Kodak PhotoCD Info via ftp

To demonstrate the use of ftp we'll log onto the Eastman KODAK Company ftp server and retrieve a file regarding PhotoCD permanence. The first step is to enter the command:

```
ftp ftp.kodak.com
```

The server responds and requests a Name and Password:

```
Connected to info.kodak.com.
220 info FTP server (Version wu-2.4(2) Fri Apr 15 08:59:57 EDT 1994) ready.
Name (ftp.kodak.com:gvalle): anonymous
331 Guest login ok, send your complete e-mail address as password.
Password:
```

Upon acceptance of the log-in information the following welcome screen is displayed:

```

230-
230-Welcome to the Eastman KODAK Company ftp server.
230-
230-* * * * *
230-* This system is for use of authorized users only.  Individuals *
230-* using this computer system without authority, or in excess of *
230-* their authority, are subject to having all their activities *
230-* on this system monitored and recorded by system personnel. *
230-* In the course of monitoring individuals improperly using this *
230-* system, or in the course of system maintenance, the activities*
230-* of authorized users may also be monitored.  Anyone using this *
230-* system expressly consents to such monitoring and is advised *
230-* that if such monitoring reveals possible criminal activity, *
230-* system personnel may provide evidence of such monitoring to *
230-* law enforcement officials. *
230-* * * * *
230-
230-
230-This system may be used 24 hours a day, 7 days a week.  The local
230-time is Sat May 20 23:06:40 1995.
230-
230-All transfers are logged with your host name and email address.
230-If you don't like this policy, please disconnect now.
230-
230-If your FTP client crashes or hangs shortly after login, try using a
230-dash (-) as the first character of your password.  This will turn off
230-the informational messages which may be confusing your ftp client.
230-
230-You are user number 52 out of a possible total of 150.
230-
230-
230-Please read the file README
230- it was last modified on Thu Feb 17 08:13:19 1994 - 457 days ago
230 Guest login ok, access restrictions apply.

```

After changing to the /pub directory we can list it's contents using the "dir" command:

```

ftp> cd pub
250-
250-All transfers from this directory are logged.
250-
250 CWD command successful.
ftp> dir
200 PORT command successful.
150 Opening ASCII mode data connection for /bin/ls.
total 1931
d----- 2 root      1          512 May 16 17:20 contrib
-rwxr-xr-x 1 root      1      119146 Feb 15  1994 gzip-1.2.4.msdos.exe
-rw-r--r-- 1 root      1      788219 Feb 15  1994 gzip-1.2.4.shar
-rw-r--r-- 1 root      1      798720 Feb 15  1994 gzip-1.2.4.tar
-rw-r--r-- 1 root      1      220623 Feb 15  1994 gzip-1.2.4.tar.gz
dr-xr-xr-x 14 root      1          512 May 18 14:11 photo-cd
d----- 2 root      1          512 Jun 27  1994 security
226 Transfer complete.
493 bytes received in 0.013 seconds (38 Kbytes/s)

```

The directory we're looking for is "photo-cd." Here's the dir listing for /photo-cd:

```
ftp> dir
200 PORT command successful.
150 Opening ASCII mode data connection for /bin/ls.
total 2805
-rw-r--r-- 1 root 1 273 Mar 9 17:12 INDEX
-rw-r--r-- 1 root 1 1553 May 16 1994 README
-rw-r--r-- 1 root 1 1438059 Mar 1 18:01 acrore.exe
-rw-r--r-- 1 root 1 1291392 Mar 1 18:01 acrore.sea
dr-xr-xr-x 2 root 1 1024 Mar 1 18:53 color
-rw-r--r-- 1 root 1 93357 May 18 14:11 descriptions.txt
dr-xr-xr-x 2 root 1 2048 May 18 14:11 general
dr-xr-xr-x 2 root 1 1024 May 10 12:12 image
lrwxrwxrwx 1 root 1 12 Feb 5 18:45 images -> r2/images
dr-xr-xr-x 2 root 1 1024 Feb 10 15:25 international
dr-xr-xr-x 2 root 1 2048 Mar 29 19:07 news
dr-xr-xr-x 2 root 1 512 May 18 1994 portfolio
dr-xr-xr-x 2 root 1 512 Nov 18 1994 prepress
dr-xr-xr-x 2 root 1 1536 May 10 12:32 printers
dr-xr-xr-x 2 root 1 1536 May 16 17:22 processors
dr-xr-xr-x 2 root 1 1536 May 10 12:38 software
-rw-r--r-- 1 root 1 4029 Oct 17 1994 survey.txt.gz
dr-xr-xr-x 2 root 1 512 Apr 13 18:03 systems
dr-xr-xr-x 2 root 1 1024 Mar 9 17:08 writable
226 Transfer complete.
1250 bytes received in 0.42 seconds (2.9 Kbytes/s)
```

But we're still not quite where we want to go. The file we need is in the /general/ directory.

```
ftp> cd general
250-Please read the file README
250- it was last modified on Thu Mar 9 10:41:08 1995 - 72 days ago
250 CWD command successful.
ftp> dir
200 PORT command successful.
150 Opening ASCII mode data connection for /bin/ls.
total 7154
-rw-r--r-- 1 root 1 0 Feb 15 1994 .notar
-rw-r--r-- 1 ftp 1 2192 Nov 29 1993 2400.txt.gz
-rw-r--r-- 1 ftp 1 1811 Nov 18 1993 6600.txt.gz
-rw-r--r-- 1 ftp 1 3600 May 18 14:11 INDEX
-rw-r--r-- 1 root 1 81 Mar 9 15:41 README
-rw-r--r-- 1 ftp 1 1690 Sep 30 1993 cdperm.txt.gz
-rw-r--r-- 1 root 1 21443 Mar 9 15:19 cdsxls.pdf
-rw-r--r-- 1 root 1 4407 May 18 14:11 descriptions.txt.gz
-rw-r--r-- 1 root 1 14240 May 15 15:11 drive.txt.gz
.
[other files]
.
-rw-r--r-- 1 ftp 1 947320 Jan 24 1994 pcddis.gif
-rw-r--r-- 1 root 1 31221 Mar 9 15:19 pcdfac.pdf
-rw-r--r-- 1 ftp 1 8144 Sep 30 1993 pcdfac.txt.gz
-rw-r--r-- 1 root 1 910 May 18 1994 pcsub.txt.gz
-rw-r--r-- 1 ftp 1 594005 Jan 24 1994 piw24.gif
-rw-r--r-- 1 ftp 1 603566 Jan 24 1994 piw42.gif
-rw-r--r-- 1 root 1 7097 May 16 17:22 piwsit.txt.gz
-rw-r--r-- 1 ftp 1 1059 Nov 29 1993 playcc.txt.gz
-rw-r--r-- 1 ftp 1 1405 Sep 30 1993 player.txt.gz
-rw-r--r-- 1 root 1 1528 Aug 29 1994 sharpe.txt.Z
-rw-r--r-- 1 ftp 1 1940 Jan 11 1994 speedy.txt.gz
226 Transfer complete.
3408 bytes received in 0.6 seconds (5.6 Kbytes/s)
```

The "get" command is used to retrieve the file `cdperm.txt`.

```
ftp> get cdperm.txt
200 PORT command successful.
150 Opening ASCII mode data connection for /bin/gunzip.
226 Transfer complete.
local: cdperm.txt remote: cdperm.txt
3733 bytes received in 0.59 seconds (6.2 Kbytes/s)
```

If we had known the location of the file in advance we could have changed to the required directory by entering the command:

```
ftp> cd /pub/photo-cd/general
```

The "bye" command is used to disconnect and exit from `ftp`:

```
ftp> bye
221 Goodbye.
```

## Additional Information

Additional information regarding `ftp` can be obtained by entering the commands:

```
man ftp
```

and

```
faq ftp
```

A demonstration of `ftp` is included in the Internet demo file `INETDM.LIT` in the VersiCOMM Updates section of AMTEC+.

## Finger

According to the `finger` documentation in the manual pages, `finger` is used to "display information about users." This is certainly true. For example, the command,

```
finger support@netcom.com
```

returns the following information:

```
[netcom]

Netcom Tech Support (support)
Home: /u33/support
Shell: /usr/local/bin/tcsh
No mail.
  User      Real Name      Idle  TTY  Host      Console Location
support  Netcom Tech Support  0:01  *q0 netcom10 (office3)

No plan.
```

Interestingly, the following query,

```
finger president@whitehouse.gov
```

returns the message:

```
[whitehouse.gov]
```

```
Finger service for arbitrary addresses on whitehouse.gov is not
supported. If you wish to send electronic mail, valid addresses are
"PRESIDENT@WHITEHOUSE.GOV", and "VICE-PRESIDENT@WHITEHOUSE.GOV".
```

If you just read the manual pages for `finger` you would never know that `finger` is also widely used to obtain news-oriented (and other) messages from sites in response to queries on particular "user" addresses. For example a query on "quake@scec.caltech.edu" will return a report on recent earthquakes in Southern California:

```
finger quake@scec.caltech.edu
```

```
[scec.gps.caltech.edu]
Login name: quake                      In real life: SCSN Data Access
Directory: /export/scec/user1/quake    Shell: /bin/csh
Last login Sun Mar 20, 1994 on tty1 from bombay.gps.calte
New mail received Sat May 20 21:51:25 1995;
unread since Fri May 19 17:24:50 1995
Plan:
WELCOME TO THE SOUTHERN CALIFORNIA EARTHQUAKE DATA CENTER (SCEC_DC)
AUTOMATED LOCATIONS OF THE CALTECH/USGS SOUTHERN CALIFORNIA SEISMIC NETWORK
```

DATE	UTC TIME	LAT.	LON.	DEPTH	MAG.	Q	COMMENT
yy-mm-dd	hh:mm:ss	(deg.)	(deg.)	(km)	typ		
95/05/15	21:59:24	34.36N	118.64W	10.9	2.5MLG	A*	6 mi. WSW of NEWHALL
95/05/17	02:22:57	34.39N	118.66W	15.0	2.9MLG	A	8 mi. SSW of CASTAIC
95/05/17	09:26:08	37.28N	118.50W	9.6	2.9MGN	A*	8 mi. SW of BISHOP
95/05/17	16:53:38	34.40N	118.66W	14.6	2.9MLG	A	8 mi. SSW of CASTAIC
95/05/17	23:27:00	36.90N	116.80W	6.0	2.7MGN	C*	31 mi. N of FURNANCE CREEK
95/05/18	10:08:42	34.09N	117.00W	5.7	2.5MLG	C	5 mi. NNE of YUCAIPA
95/05/20	14:51:44	32.90N	116.23W	11.6	2.5MGN	A*	18 mi. SSW of OCOTILLO WELLS
95/05/21	07:27:37	37.33N	117.12W	1.5	2.6MGN	B*	24 mi. NNE of SCOTTY'S CASTLE

```
-----
* Last update was at 21-MAY-1995 07:30 gmt
(List is updated when a new event occurs, not at regular intervals)

-> Entries marked with a "*" are PRELIMINARY and have NOT BEEN HUMAN REVIEWED <-
Entries may contain ERRORS and may be UPDATED or DELETED at any time.
Entries with origin times within 30 seconds may represent a single event.

This is a list of automated locations for magnitude 2.5 or greater
earthquakes recorded by the CALTECH/USGS Southern California Seismic
Network in the past 72 hours. All times are Greenwich Mean Time.
Subtract 8 hours for Pacific Standard time, and 7 hours for Pacific
Daylight time. Depths are in km.

MAGNITUDE TYPES:
MGN = empirically calibrated ML based on readings from high-gain components
MLG = ML based on synthetic Wood-Anderson response from low-gain components
ML = ML based on synthetic Wood-Anderson response from TerraScope stations

The SCEC_DC is a member of the Council of the National Seismic System.
```

The response is too lengthy to include here, but if you get a chance try:

```
finger nasanews@space.mit.edu
```

This query is included in the Internet demo file `INETDM.LIT` in the VersiCOMM Updates section of `AMTEC+`.

## Additional Information

Additional information regarding `finger` can be obtained by entering the commands:

```
man finger
```

and

```
faq finger
```

## Examples and Experiments

---



This chapter is included in order to provide a very detailed description of the use of VersiCOMM for a few specific tasks. Some examples assume the user has established an account with a particular information or communications service. All these examples assume that the default name MODEM1 is being used for the communications port. Most of these examples assume the use of a AT compatible modem.


### How To Talk To Your Modem


This example assumes you are using a modem that can be controlled via a terminal, such as a Multi-Tech MT1432 modem. Although the control of these modems can vary a great deal, most smart type modems are functionally similar. Usually these modems can operate in two different states or ways, either in ONLINE or COMMAND mode. When ONLINE the modem is simply relaying characters from your computer to the other computer and vice versa. It is said to be transparent to the user. When in the COMMAND mode instead of relaying characters it receives from you to the other computer it processes them as a command. In this way you ask the modem to dial numbers or to hang-up the phone line.

When you first turn them on most smart modems are in the COMMAND mode. If a connection is made with a modem on another computer the modem automatically goes into the ONLINE condition. Once the modem is in the ONLINE condition a special "escape sequence" must be used to return to the COMMAND mode. In AT compatible type modems this is usually a sequence of three pluses, "+++" or "+++AT". The modem must not receive any character for an interval of time termed the guard time, then receive the pluses, and then nothing else for another guard time. If this sequence is satisfied, it will escape to the COMMAND mode.

To talk to your modem, type the word COMM at AMOS command level. VersiCOMM should respond "Entering conversational mode...". Once in Conversational mode press the HOME key (or Control-^) to toggle to command mode. You should see the VersiCOMM status display. Check that the baud rate of the communications port is at a rate compatible with your modem. If the baud rate is not correct change it using the BAUD command, and then toggle back to Conversational mode (HOME key or Control-^).

Now you should be ready to talk to your modem. Because of operational quirks most AT compatible modems need to be configured so that their default settings are NOT to echo commands or send "result codes". The first command you send to your modem must (temporarily) configure it to respond with "word result codes" and to "echo" your commands. With AT compatible modems this is done by typing the command "ATQ0V1E1" followed by a . After doing this then type an "AT" followed by a . You should see the response "OK".

Other software controlled modems have similar sequences. For instance, with the Racal-Vadic VA212 Auto Dial modem the operator types a Control-E followed by a  and the dialer will respond "HELLO: I'M READY" followed by an asterisk.

To exit VersiCOMM directly from Conversational mode type a "~" character. If instead you want to exit from Command mode, toggle to Command mode using the HOME key (or Control-^ ) and type QUIT and a .

### **If You Cannot Talk To Your Modem**


If you cannot talk to your modem check that:

1. Modem is turned on.
2. Modem is connected to the correct port on your computer.
3. Modem cable is made correctly.
4. Modem is configured to echo commands and give responses.
5. The communications port is at correct baud rate.
6. Modem is in command mode.
7. The correct terminal name was specified in the VersiCOMM command line.
8. You have a modem controllable via commands from a terminal or computer.

## **How To Dial And Connect To Another Computer**

### **With a "Smart" Modem**

Assuming you have successfully been able to talk to your modem, the next step is to call another computer... If you can't find another Alpha Micro to call, check with a local computer store for the number of a computer "bulletin board." And, of course, you can always call the AMUS network computer.


At AMOS command level type COMM and then a . As before, you should see the note "Entering Conversational mode...".

If you have a AT compatible modem and have not yet generated the modem control modules using the Build function of the DIALER's Modem Setup Screen, you can dial using the following procedure. Type "AT" and verify that you get "OK" in response. Now you can dial



the number of the other computer (and modem). Type ATDT followed by the number you want to dial. For instance, ATDT5551212. The modem should then dial the number. When (and if) the remote computer's modem answers the phone and a connection is established, the word CONNECT should be displayed on your screen. Don't be concerned if the message "1200" or "2400" is also displayed.

If you have built the modem control modules for your modem (see the VersiCOMM INSTALLATION MANUAL for more info) you can use the DIAL command to dial your modem. Switch to VersiCOMM command mode and enter a command similar to the following:


```
>DIAL 5551212 
```

The DIAL command will return you to conversational mode when the host computer connection is made.

## With a Manual Modem

Using a manual modem, you dial the phone and when you hear the high pitched tone of the other modem you initiate a connection attempt using a CONNECT switch. If you successfully get a connection the CD (Carrier Detect) LED among others, should light. Many manual modems will hang-up automatically if no connection is established.

## After Establishing A Connection

Assuming the remote modem is properly connected to the remote computer, you are now a terminal on that computer. Type a few s and see if there is a response. If you called an Alpha Micro and the remote modem is connected to a port that is defined on the system and attached to a job at AMOS command level you should be able to type a command such as LOG or TIME and see a response. If you called a computer with a security login, such as the AMUS computer, it will prompt you for logon information. Bulletin boards will also ask for some info before letting you access the system.

## If No Connection is Established

If you cannot successfully use you modem to dial and connect to another computer, check that:

1. Your modem is connected to a phone line.
2. You are successfully dialing a number. Call the remote computer from another line within the same office, or confirm in some other way that you are at least dialing out.
3. There is indeed a modem and not a grimacing person answering your modem's call. If you pick up the line on which the modem is placing the call you should hear a high pitched tone. Other things you might hear, in addition to an upset voice, is continued ringing or a busy signal.
4. The remote modem is set to auto-answer.
5. The baud rate(s) and communication protocol of your modem is compatible with the remote modem. Many higher speed modems use non-standard protocols to achieve higher rates.

6. Someone on the remote system is using VersiCOMM with the modem you have called. (In this case, if everything else is correct, they would see what you type and you would see what they type.)







## How To Talk To Your Own Computer

This might seem like a strange way to start communicating with the world, but in fact it is very illustrative of the use of VersiCOMM, and gives the user control of all the elements involved. This demonstration requires that you have at least one other job on your system besides your own and that you don't mind disconnecting and connecting RS-232 cables from your computer and modem. In some instances it might also require a female to male DB-25 "gender changer."

Start by unplugging the DB-25 connector from your modem, and then unplug the DB-25 connector from the computer port that connects your other workstation (not yours) to the computer. Take the DB-25 connector you unplugged from the modem and plug it into the port that was freed by disconnecting the other workstation. With certain modems you may run into a gender problem at this point and need an adapter. This will be the cabling arrangement:

1. Your workstation will be connected to the computer as usual.
2. The communication port, instead of being connected to the modem, will be connected to a third port.
3. This third port was originally connected to a workstation other than your own.

What this arrangement does is allow you to operate two jobs from one terminal, one directly as usual, and the other using VersiCOMM.

Once the proper connections are established, type COMM and a . After entering Conversational mode toggle to Command mode and change the baud rate of the communication port to match the baud rate of the terminal that was connected to the other workstation. Often this will be 9600 or 19200 bps. Then toggle back to Conversational mode and type  a few times. If the other workstation's job was at AMOS command level when you disconnected its terminal and attached the modem cable, you should see the AMOS "dot" prompts. If you have the AMOS command level prompt type "BAUD 1200" and a  and then toggle to VersiCOMM's command mode and also type "BAUD 1200" and a . Toggle back to Conversational mode and hit  a few times. You should see the AMOS "dot" again in response to each . You are now operating as a 1200 bps terminal on the other workstation's job. If you exit VersiCOMM you are a full-speed terminal on your own workstation's job. Do a SYSTAT on the alternate job, examine the result and then exit VersiCOMM and do another SYSTAT. 1200 bps was selected for the alternate job, accessed via VersiCOMM, so you could see a difference in operation. Unless your system is heavily burdened you should be able to operate at 9600 bps. If your modem cable is configured for hardware flow control and VersiCOMM flow control has been enabled using the FLOW command, 9600 bps or higher rates will be achievable when communicating in this manner.

## Saving Displayed Information

This demonstration will assume that you have already established a connection with a remote computer. The "other computer" could be another Alpha Micro, an information or

communication service, or even your own computer as described in the previous section. For the purposes of this example another Alpha Micro will be assumed.

From VersiCOMM's Conversational mode type a command, such as LOG, to confirm that you are operating as a terminal on the other system. Once this is established toggle to VersiCOMM's Command mode and verify that the SAVE switch is OFF and then toggle back to Conversational mode. This time do a SYSTAT (not STAT) on the remote system or some other command that will result in a few lines of output from the remote system. Now key a "Control-G" and repeat the command you just executed. Most systems will display the output of the command in reduced intensity. Now toggle to Command mode and you will see that the SAVE switch is now ON. Any information displayed will also be saved in a disk file. How this file is named will be discussed in a moment. Toggle back to Conversational mode. You can use "Control-G" to repeatedly turn the SAVE switch on and off, saving some information but not other. When you are done exit VersiCOMM.

Because no filename was specified for the SAVE, VersiCOMM assumed a default of the communications port's terminal-name with an extension of LST. In this case the file is MODEM1.LST. You can VUE or type this file to verify its content.

### Naming A Save File

What if you want to save information in a particular file, not MODEM1.LST? No problem, instead of keying "Control-G" to initiate the save, toggle to Command mode and use the SAVE command. At the command prompt type SAVE followed by the name of the file in which you want to save the displayed information. If a SAVE file is already opened it must first be closed using the CLOSE command. Note that multiple output files can be created by using the CLOSE command to close an existing file, and then SAVE to open another.







When saving information from information services, occasionally some networks and services will inject nulls into the text. This is not noticeable in the display of the text, but if you VUE a file with embedded nulls it will APPEAR that text has been lost. Actually its there, but VUE is ignoring it. If you type the file the text can be seen. This can be avoided by filtering any nulls from the remote computer. This is done automatically if translation or filtering of any other remote character is being done.



### Translating & Filtering Characters

VersiCOMM has two independent translation and filtering tables, one for remote characters, i.e. characters received at the communications port from the remote computer, and one for local characters, i.e. those characters that originate at the local system such as characters typed at the keyboard or sent in a USING file or a conventional file to the remote system. You may want to review the operation of the LOCAL, REMOTE, TLOCAL, TREMOTE and FLUSH commands before continuing with this demo.

Once again in this demonstration we assume that you are "talking to" another Alpha Micro and to start you are in VersiCOMM's Conversational mode.

To begin, run SYSTAT on the remote Alpha. When it is complete toggle to VersiCOMM's command mode and type "TLOCAL" and a , then type "LOCAL 65,66" and a . Now go back to Conversational mode and run SYSTAT on the remote computer. What happened? The remote computer should have echoed "SYSTBT" as you typed "SYSTAT" and then responded with something like "?SYSTBT?". Any "A" was translated into a "B" BEFORE it was sent to the remote computer.

Now, go to Command mode and type "TREMOTE" and a  and then "REMOTE 66,65" and a  and then return to Conversational mode. Run SYSTAT again and see what happens. The command is echoed "properly" but the system doesn't understand it. It responds "?SYSTAT?". This is because it actually received the sequence "SYSTBT" and as it echoed it back to your computer the "B" was translated into an "A".

Now, return to Command mode and type "FLUSH" and a . This will clear the translations that have been specified. Then type "REMOTE 65,0" and a  and go back to conversational mode. Run SYSTAT again. As you type "SYSTAT" it will be echoed "SYSTT" and any "A" will be filtered from the resulting output.



**APPLICATION NOTE**  
**TURNING TRANSLATION TEMPORARILY OFF**

If you want to temporarily disable translation and filtering it is not necessary to FLUSH the tables and rebuild them later. You can switch translation and filtering off and on using the TLOCAL and TREMOTE commands.



**APPLICATION NOTE**  
**TRANSLATION DURING FILE TRANSFERS**

When transferring files between systems keep in mind that any translation or filtering in effect will also apply to characters transmitted via the SEND command or received via the RECEIVE command. You will normally want to turn any translation or filtering OFF when transferring files using these commands.



**APPLICATION NOTE**  
**FILTERING NULLS**


The filtering of a null (ASCII 0) is a special case. Nulls are automatically filtered if translation is enabled and a translation table exists. A translation table is not allocated unless a translation (or filtering) is specified. Usually when nulls are to be filtered, other characters are to be filtered as well, so the table will have been allocated. If you want to filter only nulls, then the table must be allocated by executing a REMOTE command, and then FLUSHed. This will leave the table allocated.

## How To Write A Using File

Included with VersiCOMM, or provided as examples elsewhere in this manual are sample USING files for accessing EasyLink, MCI Mail and Dow Jones/News Retrieval. In this example the process involved in writing a USING file is discussed as a USING file to access the Compuserve Information Service is written. In order to be able to test and use the resulting USING file you must subscribe to Compuserve and have valid logon information. Even if you do not plan to access Compuserve you may want to review this example, since the procedure is similar for accessing other services, and the use of several important VersiCOMM commands is described.

The first thing to do when you are going to write a USING file is to manually "walk through" the entire dialog which the USING file will be automating. In this case, we will go through the steps in using VersiCOMM to access Compuserve via a local Compuserve Network access number using a AT compatible smart modem.

The first step would be to run VersiCOMM and after entering Conversational mode, we might toggle to Command mode and make sure the baud rate of the communications port was set to 2400 or 9600 bps, depending on the access number used and the capabilities of the modem. After setting the baud rate we would switch back to conversational mode.

Next, the modem would be commanded to dial Compuserve's access number by typing the command line "ATDTxxxxxx" and a  or switching to VersiCOMM command mode and entering a DIAL command.

After dialing the number and establishing a connection the modem would respond with the message "CONNECT 2400" or "CONNECT 9600".

This is a critical point in automating access to a service. Some services respond with a message once they detect a successful connection, others wait for something to be typed by you before they respond. Compuserve will respond with the "User ID:" prompt if you send a ^C (ASCII 3).

In response to the prompt you enter your User ID. Compuserve then responds with "Password:".

After the password is entered, Compuserve will respond with logon data and its "top" Menu.

VUE is normally used to write a USING file. For this demo, name your USING file "COMP.TLK".

If you are writing a USING file and have yet to debug it, the first command should always be the DIALOG command. This will cause VersiCOMM to display the dialog between itself and the remote computer as it attempts to process your using file. When specifying a VersiCOMM command in a USING file don't forget to precede it with a "\$". In a USING file any line that is not preceded by a \$ is sent to the communications port, and from there to your modem and on to the remote system.

In most cases, when communicating with an online service using ASCII characters, STRIP should be turned ON.

Based on our "walk through", next we should set the communications port baud rate to 2400 or 9600 bps. This is done using the BAUD command. Then the Compuserve number is dialed and we wait for a "CONNECT" message before continuing. If the "CONNECT" message is received, we are "online" with Compuserve and may continue with the logon procedure. If no "CONNECT" is received we will exit VersiCOMM.



The way a USING file "waits" for a particular response from a remote computer is by specifying the string in a WAIT command. If the string is received from the host the line immediately following the WAIT command line is skipped, otherwise it is processed. In this case if no connection is made the line following the WAIT will be a "GOTO 100", which will skip to a part of the USING file that prints a message and quits. If the connection is successful the "GOTO 100" line is skipped and a "GOTO 200" is executed instead. Line 200 will be the start of the logon sequence. Here is what has been written so far:

```



$DIALOG
$STRIP ON
$BAUD 2400
ATDTxxxxxxx
$WAIT CONNECT
$GOTO 100 ;No Connect
$GOTO 200 ;Have Connect
;
;No Connect Exit
;100
$PRINT !! Did not CONNECT to Compuserve !!
$QUIT
;
;Log On Sequence Starts Here
;200

```

As is the case with most programming, it's usually not a good idea to try to write the entire USING file in one fell swoop. It's better to start with a small segment, make sure it works, and then build on it in steps until it does all you want it to do. This is a good point to stop to test this USING file and see if we can successfully make a connection.


After saving the USING file, at AMOS command level type "COMM USING COMP". If all goes well you should find yourself connected to the Compuserve Network, and if you key a  it should respond with "User ID:". YOU MUST KEY  TO GET THE CORRECT RESPONSE.





If this did not happen chances are VersiCOMM timed out and you saw the message that said "!!Did not CONNECT to Compuserve". Unless the wait time is changed using the WAITIME command, VersiCOMM will wait about 40 seconds for a character in a WAIT string to be received. If your USING file did not establish a connection, the Compuserve access number may have been busy, or for some reason not answered. Other problems could be that your modem is not connected to the phone line, or it is not responding with words result codes.

If you are successfully connected to the Compuserve Network, instead of continuing with the logon procedure, hang-up the modem. With AT compatible modems this can be accomplished by pausing a second, then typing the character string "+++" and then NOT typing a  or anything else. After a moment, your modem should respond with "OK". This means you're back in command mode and can enter the command "ATH0" and a . This should hang-up the modem.

Now, back to writing the rest of the USING file. After connecting to the Compuserve Network the next step is to send the network a 'Control-C', correct? CAUTION!! Experience has shown that a modem can be a deceiving beast. With some modems I have experienced a situation where, if a character is sent IMMEDIATELY upon receipt of the modem's CONNECT message, the modem will not handle it properly. The modem may require a brief pause to establish a usable connection. So, as a precaution, I normally follow the wait for the CONNECT message with a pause of one or two seconds. After you have a working USING file you could vary this parameter to see if your modem requires it.

In order for Compuserve to give us the "User ID:" prompt we then need to send it a 'Control-C'. This is accomplished by making the first character on the line following the PAUSE command a 'Control-C'. To enter a "^C", ASCII 3, follow this procedure:

1. Key <ESCAPE> to switch to VUE command mode.
2. Type CONTROL TRUE and a .

3. Key <ESCAPE> to return to VUE text entry mode.
4. Position the cursor where you want the ^C to be entered.
5. Key  and  at the same time. No character will be typed.
6. Key  and  at the same time. A "^C" will be typed.

After sending Compuserve the "^C" then we need to wait for the "User ID:" prompt. Rather than waiting for the entire string we wait for a significant part of it, such as "ID:". In this case the WAIT command is on the same line with the "^C" so only a "^C" will be sent to Compuserve and not a "^C" followed by a carriage-return. Then in response to the prompt we send your user ID. In this example this is "password1". Next, we wait for Compuserve to respond with the "Password:" prompt, and then send the password, designated here as "password2".

VersiCOMM will continue to process a USING file until one of three conditions is encountered. If a TALKTO command is processed, control of VersiCOMM by the USING file is released and the user is put in Conversational mode. If the end of the USING file is reached, VersiCOMM proceeds as if a TALKTO command had been processed. If VersiCOMM executes a "QUIT" command it looks to see if a filespec was specified, if not it looks to see if the file "trmdef.HU0" (or "HANGUP.TLK") exists in the current account. If it does, it is processed as a USING file and sent to the host before exiting.

In this case a TALKTO command terminates normal execution of the USING file and returns control to the user. If either of the logon prompts is not received as expected, the USING file branches to an error exit terminated by a QUIT.

Your completed USING file should be similar to the following:

```
$DIALOG
$STRIP ON
$BAUD 1200
ATDTxxxxxxx
$WAIT CONNECT
$GOTO 100 ;No Connect
$GOTO 200 ;Have Connect
;
;No Connect Exit
;100
$PRINT !! Did not CONNECT to Compuserve !!
$QUIT
;
;Log On Sequence Starts Here
;200
$PAUSE 30
^C$WAIT ID:
$GOTO 300
password1
$WAIT word:
$GOTO 300
password2
$TALKTO
;300
$PRINT !! Error in Compuserve Logon !!
$QUIT
```

Run VersiCOMM again USING this file. It should complete the access to Compuserve. If not, carefully review the file and be sure everything is typed correctly. You have already logged on manually, so you know your passwords are valid.

Once you have a USING file that successfully accesses a service, the next step is to note, as you use the service, any customizing you need in the operation of VersiCOMM. For instance, the Compuserve computer uses a backspace rather than <DEL> for correcting typed text, so it is convenient to use VersiCOMM's translation facility to translate any <DEL> to a <BS> before sending it to the Compuserve computer. The using file commands to do this are:

```
$TLOCAL
$LOCAL 127,8
```

A more complete USING file for accessing Compuserve, CIS.TLK, will be found in release account [7,121].

## How To Send An MCI Letter


MCI Mail and AT&T's EasyLink are two popular mail services. Each service has its advantages. For general correspondence I prefer MCI Mail. Letters are printed in letter quality, upper and lower case text on white bond paper with a laser printer. You may optionally "register" your letterhead and signature with MCI Mail and these can be reproduced on your MCI letters as well. One big advantage over EasyLink is that you can check the letter for transmission errors BEFORE it is sent to the addressee.




This demonstration requires you have an account with MCI Mail. At the time of this writing there is no charge for an account or a monthly minimum charge or any connect time charges for a basic user.

This is a brief description of how to send an MCI Letter. For detailed information refer to the documentation you received from MCI, or type "HELP" at any MCI Mail menu.


First, use VUE to create the text of the letter you wish to send. Do not include address information in this file. This will be entered separately. We'll name this letter "LETTER.TXT".

Access MCI, either manually or automatically with a USING file, and select CREATE from the Main Menu.


At the "TO:" prompt enter the first line of the address, i.e. "Mr. John Doe", or "ALPHA MICRO", and then a . MCI will search its customer list for this name. If it finds this name, MCI will assume you wish to send this message to the addressee's electronic mail box and not by surface mail. If want to override this choice, when you type the first line follow the addressee's name with the option "(ADDRESS)", i.e. you might enter "Mr. John Doe (ADDRESS)". This will tell MCI Mail that you want to enter a postal address, even if the person has an electronic mail box.

At this point you will be prompted to enter the remainder of the address. Do this, and then on the last line just type a . This tells MCI the address is complete. MCI then prompts you with another "TO:". If you don't want to send this letter to more than one address, just hit . MCI then prompts "CC:". This is for "courtesy copies" to other addresses. Key a  if none





are desired. MCI will then prompt you with "Subject:". You may enter a brief note or just a .

MCI now prompts you for the text of the letter and tells you to type a "/" on a line by itself to signify the end of the text. At this point you can enter text manually or upload a prepared text file using any of several protocols. In this example we will upload LETTER.TXT using the ZMODEM protocol. Starting in the first column and on the first line enter the text:

```
/UPLOAD ASCII ZMODEM 
```

When MCI prompts you to upload the file, switch to VersiCOMM Command mode and enter the command:

```
SENDZ LETTER.TXT 
```

When the transmission is complete, return to Conversational mode. If all of the "handling options" menu is not displayed, key a  to redisplay the menu.

This menu displays the various things that you can do with your "DRAFT" letter. You may want to use the READ PAPER option to see how it will appear when printed out on paper. This is especially true if the letter is more than one page long.

If you are satisfied with the appearance of the letter, the manner in which it is to be delivered can now be selected. "SEND" will send the letter electronically to an MCI postal center nearest the addressee, and then deliver it via U.S. Mail. This often results in next day delivery. "SEND ONITE" is handled in the same manner except it is delivered by courier and delivery is by noon the next day. A four hour delivery option also is available when sending letters to many metropolitan areas.

MCI Mail can also be used to send Telexes. Rather than entering a postal address you enter the Telex address. Type HELP TELEX at the main menu for details.

## How To Send Batch EasyLink Messages

This demonstration requires that you have established an EasyLink account.

Refer to EasyLink documentation for complete details on sending Mailgrams, Telexes and other messages.

Use VUE to create a file "BATCH.TXT". The general form of the file should be as follows:

```
/BATCH
62999999+
THIS IS THE TEXT OF THE FIRST MESSAGE
LLLL
/ZIP
JOHN DOE
123 MAIN STREET
MIDVILLE NY 10000,
/ZIP
MARY DOE
345 MAIN STREET
MIDVILLE CA 90000+
THIS IS THE SECOND MESSAGE
LLLL
/WUW 8516543+
THIS IS THE THIRD AND FINAL MESSAGE
MMMM
```

Note particularly these conventions:

Most messages are limited to a line width of 68 characters.

A comma is used only for separating addresses when a message is being sent to multiple addresses. AN INDIVIDUAL ADDRESS MUST NOT CONTAIN ANY COMMAS.

A plus terminates the address field of a message.

"LLLL" signifies the end of text for a message and tells EasyLink to send the message and stay online.

"MMMM" signifies the end of text for a message and tells EasyLink to send the message and disconnect.

In this batch the first message is sent to the EasyLink electronic mailbox address 62999999. The second message is sent as a mailgram to the two addresses listed. The third message is sent to a Worldwide Telex Number.

To send the batch of messages, access EasyLink and when the "PTS" prompt is received toggle to command mode. Check first that the LINEFEED switch is set to OFF and then use the SEND command to send the file "BATCH.TXT". When the transmission is complete and the messages have been accepted you will be disconnected from EasyLink.

Note that it would not be difficult to write a USING file that automatically accesses EasyLink, and sends a batch of messages.

## Accessing The AMUS Network Computer


After you have used the Access Parameter Definition Program in [7,121] to define your AMUS access number and logon information, the COMMCENTRE function "Access AMUS Computer" can be used to dial and logon to the AMUS computer. If you are an AMUS member and do not know your access codes, contact the AMUS office to obtain this information.

## Transferring Files To Or From AMUS


I. Using the VersiCOMM-Plus commands TRANS or RETRV

TRANS and RETRV may be used to transfer any type of AMOS file to or from AMUS with CRC16 error detection and correction.

To TRANSMIT a file TO AMUS:

1. Log into the AMUS account to which the file should be transferred.
2. Use the HOME key (or Control ^) to toggle to VersiCOMM command mode.
3. At the command prompt type "TRANS filespec ", where filespec is the file specification of the file you wish to transfer to AMUS.
4. When TRANS has completed transmission of the file, use the HOME key (or Control-^) to return to conversational mode.




To RETRIEVE a file FROM AMUS

1. Log into the AMUS account from which the file should be transferred.
2. Use the HOME key (or Control-^) to toggle to VersiCOMM command mode.
3. At the command prompt type "RETRV filespec ", where filespec is the file specification of the file you wish to retrieve from AMUS.
4. When RETRV has completed receipt of the file, use the HOME key (or Control-^) to return to conversational mode.

## II. Using the VersiCOMM commands SEND or RECEIVE

SEND and RECEIVE can be use to transfer sequential files, including binary" files such as LIT, PRG and RUN files. No error checks are done.

To TRANSMIT a file TO AMUS

1. Log into the AMUS account to which the file should be transferred.
2. At the AMUS "dot" type "VREC filespec ", where filespec is the name of the file you wish to transfer to AMUS. The message "Will wait 60 seconds for you to SEND filespec" will be displayed.
3. Use the HOME key (or Control-^) to toggle to VersiCOMM command mode.
4. Check and be sure the LINEFEED switch is ON. If it is not ON then at the VersiCOMM command prompt type "LINEFEED ON .
5. At the VersiCOMM command prompt type "SEND filespec ", where filespec is the file specification of the file you wish to transfer to AMUS.
6. Wait for SEND to complete transmission of the file.

7. Later versions of VersiCOMM will automatically return you to conversational mode when SEND is done, otherwise use the HOME key (or Control-^) to return to conversational mode. Before typing anything in conversational mode, wait a few seconds for the message "filespec RECEIVED AND CLOSED" to be displayed.
8. Type "DIR/H filespec␣" to obtain the hash code of the received file.
9. Return to VersiCOMM command mode and type "AMOS DIR/H filespec ␣␣", to check if the hash code matches that of the received file.
10. Return to VersiCOMM conversational mode.

#### To RETRIEVE a file FROM AMUS

1. Log into the AMUS account from which the file should be transferred.
2. At the AMUS "dot" type "VSEND filespec␣", where filespec is the name of the file you wish to transfer from AMUS. The message "Will wait 20 seconds for you to RECEIVE filespec" will be displayed.
3. Use the HOME key (or Control-^) to toggle to VersiCOMM command mode.
4. At the prompt type "RECEIVE filespec␣", where filespec is the file specification of the file you wish to receive from AMUS.
5. When the receipt of the file has been completed, use the HOME key (or Control-^) to return to conversational mode.
6. Type "DIR/H filespec␣" to obtain the hash code of the AMUS file.
7. Return to VersiCOMM command mode and type "AMOS DIR/H filespec ␣␣", to check the hash code of the file as it was received on your system.
8. Return to VersiCOMM conversational mode.

## How To Use DJNS To Get Quotes And News

This demonstration assumes you have established an account with Dow Jones News/Retrieval. If you subscribe to MCI Mail you should also have access to DJNS. Note that connect time and other charges apply when you are using the services of DJNS.

Using VersiCOMM, access DJNS. After a successful connection has been established the Main Menu of databases will be displayed.

The first step in retrieving info about a company is to determine its company code. If you do not know the company code type //SYMBOL at a selection prompt. This results in access to a database of company codes. For this demo we will find the company code for Alpha Micro. At the SYMBOL menu, select "STOCK SYMBOLS" by typing a "1" and a ␣. You will be prompted to enter a query. Enter "ALPHA MICRO" and a ␣. The company code "ALMI" will be returned. This symbol will be used in all examples which follow.

To get the most recent headlines on a company type ".ALMI 01" and a ↵.

To get the story associated with a headline type the two letter story identifier code (listed with the headline) and a ↵.

To get the latest news story on a company type ".ALMI" and a ↵.

To get a current stock quote type ",ALMI" and a ↵.

To get daily historical quotes for the past 24 days type ";ALMI P1" for days 1 to 12 and ";ALMI P2" for days 13 to 24. Don't forget the ↵.


DJNS has a free user "newsletter". This is accessed by typing //INTRO. General user information and news topics are listed in the //INTRO menu. You can also type //DJ HELP for an overview of the use of the system.



# Errors

---

## **?A SAVE file has already been specified**

You must CLOSE the currently open SAVE file before you can use the SAVE command to open a new output file. At the VersiCOMM Command mode prompt enter "CLOSE" followed by a , and then try again.

## **?All communications ports are in use**

No communication port is currently available for use. All modem ports are busy with local or remote users. A "STAT" command can be entered at the AMOS dot prompt to check the status of each user on the system.

## **?AMOS xxxxx?**

The AMOS command specified could not be found. Check the spelling of the command and try again.

## **?An emulation has already been specified**

Before specifying a new emulation the current emulation must first be turned off by entering the command "EMULATE OFF".

## **Argument number may only be in the range 0 - 15**

An invalid argument number was specified in an ARG or FLG command.

## **?Assigned FAX job not running FAX program**

The FAX job was not running FAXSER or other FAXxxx program.

## **?Baud rate specified is not valid**

The baud rate requested was incorrectly entered or is not supported by the hardware interface.

**?Bypassing USING file -- nest limit exceeded**

Each time a script file is executed from within a script file, information is saved in a structure called a stack. Excessive nesting will exhaust the stack space that is available, resulting in this error.

**?Can only FLUSH LOCAL or REMOTE**

A 'FLUSH' command was specified with an argument other than LOCAL or REMOTE.

**?Cannot find DSK0:BAUD.xxx[1,4] or MEM:BAUD.xxx**

In pre-1.3B and 1.0D systems, VersiCOMM uses the system program BAUD.LIT to perform its baud changing chores and cannot locate the program.

**?Cannot use this driver -- impure area overflow**

The driver specified in a CHGTDV command has an impure area requirement larger than the driver that is currently in use.

**?Cannot use without RCLOCK implemented in SYSTEM**

The JWAIT, FLOW, IDLEOUT, RECZ and SENDZ commands require that RCLOCK has been implemented in SYSTEM memory. See the VersiCOMM INSTALLATION MANUAL for information on RCLOCK.

**?Character value can only be 0 to 255**

Translation codes and several other character codes may only be in the range of 0 to 255.

**?Command requires AMOS32 1.0 or AMOSL 1.3B or later**

Several commands require these or later versions of the AMOS operating system.

**?Data SIZE may only be 5 to 8 bits (if supported)**

You may only specify an argument of 5, 6, 7 or 8 in the SIZE command, and then only if the hardware supports the size specified.

**?DTR and RTS not supported on this interface**

Either the 'DTR' or 'DTRLOW' command was specified and the modem port does not support either DTR or RTS.

**?DTR not supported--will try RTS...**

If either the 'DTR' or 'DTRLOW' command is specified and the modem port does not support DTR, VersiCOMM will try to use RTS instead.

**?FAX job not in CMD or COM state**

The FAX job was not running FAXSER, FAXCMD or FAXCOM.

**?FLOW can only be ON, OFF, HW ON, HW OFF, SW ON or SW OFF**

An invalid argument was specified for the FLOW command.



**?Illegal or unsupported port configuration**

Not all combinations of values for PARITY, SIZE and STOP are valid or supported.

**?Interface not supported -- requires RTS on pin 5**

'FLOW ON' or 'FLOW HW ON' was specified and the port hardware or software does not support RTS. By Alpha Micro conventions RTS is an output usually found on pin 5, however it is NOT required to be on pin 5.

**?Keyword not found**

A keyword other than "USING" was specified in the initial .COMM command line

**?Maximum packet size is 64**

The maximum packet size that can be specified in a SENDEPX command is 64.

**?No SAVE file is currently active**

You attempted to CLOSE a SAVE file that was not open.

**?No data received**

No data was received from the remote computer or device.

**?No echo detected -- will skip EPX checks**

The SENDEPX file will skip echoplex error checking if no echo is detected.

**?No emulation has been specified**

An emulation must be in effect for the EMULATE OFF command to be used.

**?Number of STOP bits may only be 1 or 2**

You may only specify 1 or 2 as an argument in the STOP command.

**?PIC file read error**

The file COMM.PIC or COMMR.PIC was not found in DSK0:[7,0].

**?Parity may only be NONE, ODD, or EVEN**

These are the supported values for parity.

**%Possible error detected in file transfer%**

A received character did not match the value that was expected to be echoed.

**?Phone number file not found -- ^C to abort**

The file specified in a DIAL command could not be found.

**?RC.LOK set for more than 20 seconds**

RC.LOK is a semaphore used to mediate the assignment of a modem and should never be locked for an extended period of time. You will need to reset the system to clear the lock. Notify Valley Programming if this frequently occurs.

**?RCLOCK.SYS in system memory but not running**

RCLOCK.SYS was not started by an RCLOCK command following the last SYSTEM command in the system initialization command file.

**?Requires RCLOCK.SYS (101) or later**

A 'FLOW ON' command was entered and the version of RCLOCK.SYS loaded into system memory was from an earlier release of VersiCOMM.

**?RTS and DTR not supported on this interface**

Either the 'RTS' or 'RTSLOW' command was specified and the modem port does not support either RTS or DTR.

**?RTS not supported--will try DTR...**

If either the 'RTS' or 'RTSLOW' command is specified and the modem port does not support RTS, VersiCOMM will try to use DTR instead.

**?SAVE file must be CLOSED before executing RECEIVE**

You cannot RECEIVE a file with a SAVE file open.

**?Soft-Parity may only be NONE, STRIP, EVEN or ODD**

These are the only supported values of soft-parity.

**?Switch may only be ON or OFF**

An argument was specified other than ON or OFF.

**?Terminal not found**

The terminal specified, or the default terminal MODEM1, was not defined in your system as a terminal.

**?Terminal driver not found**

The terminal driver specified in a CHGTDV command could not be found in the terminal definition chain or system memory.

**?Terminal does not support a bottom status line**

The command 'STATUS ON' was specified and your terminal does not support an addressable bottom status line.

**?Terminal does not support save screen area**

The command 'PRESERVE ON' was specified and your terminal does not support the save screen area function.

**?Throttle not effective in pre-AMOS 1.2 mode**

If operating under a pre-AMOS 1.2 system, a throttle parameter cannot be specified in a SEND command.

**?Timed out waiting for prompt character**

The SENDL commands waits for a prompt character defined by the PROMPTCHR command before sending the next line of data. The default prompt character is a colon.

**?Translation specification error**

An character code outside the range of 0 to 255 was specified.

**?What**

VersiCOMM could not interpret the command specified.

**?You cannot specify your own terminal**

You cannot specify your own terminal as the communications terminal.



# Index

---

## 8

8-bit character codes, 123

## A

ADDFLG, 131

AMOS, 132

AMOS COMMAND, 39

AMOS Command

    Overview, 17

AMUS Network

    Downloading Files, 212

ANDFLG, 132

ANSWERBACK, 132

APPLICATION NOTE

    Filtering Nulls, 206

    Timing in USING Files, 65

    Translating/Filtering High Codes, 39

    Translation During File Transfer, 206

    Turning Translation Off, 206

    Using \$ in Logon, 64

    VPS Communication SBRs, 69

    Wait Character Sequence, 64

ARGn

    Replacing with @n, 120

    Setting the Value of, 120

ARGn and FLGn

    List of Commands, 121

ATTENTION, 132

AUTOLF, 133

AutoLF

    In Status Display, 28

Automatic Timeout Exit, 27

## B

- BACKTO, 133
- BASIC Interface, 111
  - ARGn and FLGn Values, 120
  - GCMARG, 112
  - GETPIN, 118
  - PCMARG, 112
  - Required Map, 111
  - SETPIN, 119
  - TREAD, 112
  - TWAIT, 116
  - TWRITE, 112
- BAUD, 133
- Baud Matching, 41
- Baud Rate
  - In Status Display, 30
- BAUTO, 133
- BPXFER, 134
- BREAK, 134
  - Description of, 27
- BREAK Generation
  - In Conversational Mode, 27
- BREAKCHR, 134
- BreakChr
  - In Status Display, 31

## C

- CABORT, 134, 135
- CCENDTLK, 135
- CCENDTLK Command
  - Use in USING File, 66
- Changing Terminal Drivers, 44
- CHGMODE, 135
- ChgMode
  - In Status Display, 31
- CHGTDV, 44
  - Errors, 45
  - Operation, 44
- CLEAR, 135
- CLOSE, 135
- COMM LOG File
  - Overview, 16
- COMMAND Character, 136
- Command File
  - To Run VersiCOMM, 39
- Command Format
  - Description, 21
- Command Mode
  - Operation In, 28
  - VersiCOMMander, 32
- Command Mode Status Display

Example, 28

Commands

- ADDFLG, 131
- AMOS, 132
- ANDFLG, 132
- ANSWERBACK, 132
- ATTENTION, 132
- AUTOLF, 133
- BACKTO, 133
- BAUD, 133
- BAUTO, 133
- BPXFER, 134
- BREAK, 134
- BREAKCHR, 134
- CABORT, 134, 135
- CCENDTLK, 135
- CHGMODE, 135
- CLEAR, 135
- CLOSE, 135
- COMMAND, 136
- COMMENT, 136
- CSTATUS, 135
- CTRLC, 136
- DECFLG, 136
- DIAL, 137
- DIALOG, 137
- DIREXIT, 137
- DIRSAVE, 138
- DISPLAY, 138
- DOSCRIP, 137
- DOTCRT, 137
- DTR, 138
- DTRHANGUP, 138
- DTRLOW, 139
- ECHO, 139
- EMULATE, 139
- EOLCHR, 140
- EOLTIME, 140
- ESCAPE, 140
- EXIT, 141
- FKEY, 141
- FLOW, 141
- FLUSH, 142
- GOTO, 142
- HANDSHAKE, 142
- HANGUP, 142
- HELP, 142
- IDLEOUT, 143
- INCFLG, 143
- INPARG, 143
- INPINC, 144
- INUSE, 144

JWAIT, 144  
LABELCHR, 144  
LINEFEED, 144  
LISTEN, 144  
LOCAL, 144  
LOOKUP, 145  
MODEM, 145  
MOVCUR, 146  
MVER, 146  
NEWLINE, 146  
NOTELOG, 146  
NOTESAVE, 146  
ORFLG, 146  
PARAMCHR, 147  
PARITY, 147  
PAUSE, 147  
PCHR, 148  
PRESERVE, 148  
PRINT, 148  
PROMPTCHR, 148  
PTIME, 148  
PWRITECR, 148  
QUIT, 148  
RECEIVE, 149  
RECORD, 149  
RECX, 149  
RE CZ, 149  
REDIAL, 151  
REMOTE, 151  
RETRV, 151  
RTS, 152  
RTSLOW, 152  
RUN, 152  
SAVE, 153  
SELECT, 153  
SEND, 153  
SENDEPX, 154  
SENDL, 154  
SENDTLK, 155  
SENDX, 155  
SENDZ, 155  
SETARG, 156  
SETFLG, 156  
SIZE, 157  
SPARITY, 157  
STATUS, 157  
STOP, 157  
STRIP, 158  
SUBFLG, 158  
TALKTO, 158  
TALL, 158  
TLOCAL, 158



- TRANS, 158
- TREMOTE, 159
- TRMFLG, 159
- TSTARG, 160
- TSTDIALOG, 160
- TSTFLG, 160
- TYPE, 161
- TYPECR, 161
- TYPEFLG, 161
- TYPEWAIT, 161
- WAIT, 161
- WAITIME, 162
- WASIT, 162
- WPAUSE, 163
- WPAUSECHR, 163
- WPREFIX, 163
- WPREFIXCHR, 163
- WRAP, 163
- WRITE, 163
- WRITECR, 164
- X command, 164
- XORFLG, 164
- ZAUTO, 165
- ZDLE, 165
- COMMENT, 136
- Compuserve
  - Access vis DIALER, 57
  - Writing a USING file for, 206
- Conditional Branching
  - In USING File, 64
- Connecting to Another Computer, 202
- Control Characters
  - How to Enter in VUE, 208
- Control Keys
  - In Status Display, 31
- CONTROL-@
  - Use in Conversational Mode, 27
- CONTROL-^
  - Use in Conversational Mode, 27
- CONTROL-G
  - Use in Conversational Mode, 26
- Conversational Mode
  - Operation In, 25
- CSTATUS, 135
- CTRLC, 136
- CTRLC Command
  - Use in USING File, 65

## D

- DECFLG, 136
- Default Operation
  - Changing, 24

- DIAL, 137
- DIALER, 47
  - Accessing CIS, 57
  - AlphaBASIC Source File, 47
  - Data Files, 47
  - DIALER Screen, 48
  - Entry Screen, 50
  - Example Entry, 57
  - Function Key Operation, 48
  - Overview, 47
  - Subroutines, 48
- DIALER Entry Screen, 50
  - Field Editing, 53
  - Fields, 50
  - Function Key Actions, 53
- DIALER Modem Setup Screen
  - Field Editing, 56
  - Fields, 54
  - Function Key Actions, 56
- DIALER Screen, 48
  - Function Key Actions, 49
  - Selection Controls, 48
- Dialing Another Computer, 202
- Dialing Methods, 40
  - Conversational Mode, 40
  - DIAL Command, 41
  - Script File, 41
  - The COMM CENTRE, 40
  - The DIALER, 41
- DIALOG, 137
- DIALOG Command
  - Use in USING File, 66
- DIREXIT, 137
- DirExit
  - In Status Display, 31
- DIRSAVE, 138
- DirSave
  - In Status Display, 31
- DISPLAY, 138
- DOSCRIP, 137
- DOTCRT, 137
- DOW JONES
  - Quotes and News, 214
- Driver
  - In Status Display, 29
- DTR, 138
  - In Status Display, 30
- DTRHANGUP, 138
- DTRLOW, 139

## E

- EasyLink

- Batch Messages, 211
  - USING File to Access, 62
- EasyLink QuikLetter
  - Overview, 18
- ECHO, 139
- Echo
  - In Status Display, 28
- Echoplex Error Detection, 154
  - Used with SEND, 73
- Electronic Mail, 186
- elm
  - Default editor, 187
  - Documentation, 188
  - Getting Help, 187
  - Reading a message, 188
  - Replying to a message, 188
  - Saving a message, 188
  - Sending a message, 187
- Email, 186
  - elm documentation, 188
  - Reading a message, 188
  - Replying to a message, 188
  - Saving a message, 188
  - Sending a message, 187
- EMULATE, 139
- Emulate
  - In Status Display, 29
- EOLCHR, 140
- EOLChr
  - In Status Display, 31
- EOLTIME, 140
- EOLTime
  - In Status Display, 31
- ESCAPE, 140
- EXIT, 141
- Exiting VersiCOMM
  - From Conversational Mode, 27
- Extended Function IDVs
  - Overview, 18

## F

- File List Processing, 78
- FILTER, 45
- Filtering a SAVE File, 45
- Filtering Characters
  - Overview, 17
- Filtering Nulls, 205
  - Special Case of, 39
- finger, 198
  - Additional Information, 200
  - Earthquake information, 199
  - NASA news, 199

FKEY, 141  
FLGn  
    Setting the Value of, 120  
FLOW, 141  
FLUSH, 142  
ftp, 194  
    Additional Information, 198  
    Commands, 195  
    Example session, 195  
    Starting, 195

## G

GCMARG XCALL, 112  
GENERIC TERMINAL EMULATION, 126  
Generic Terminal Emulation  
    Overview, 18  
GETPIN XCALL, 118  
gopher, 182  
    Additional Information, 186  
    Keyboard Controls, 183  
    Searching Using Veronica, 184  
    Starting, 182  
    Terminal Notes, 182  
    Too many connections, 184  
gopher example  
    Technical Service Bulletins, 185  
    Weather data, 185  
GOTO, 142

## H

HANDSHAKE, 142  
Handshake  
    In Status Display, 30  
HANGUP, 142  
HANGUP.TLK  
    Special Purpose USING File, 68  
    To Hangup Smart Modem, 68  
HELP, 142  
HELP System  
    Overview, 18

## I

IDLEOUT, 143  
Idleout Exit  
    Overview, 19  
In Use Bit  
    Clearing in an Emergency, 43  
INCFLG, 143  
Information Retrieval  
    Automated, 67  
INI.TIW File

- Description and Use, 23
  - Overview, 16
- INI.TLK
  - Description and Use, 24
- INI.TLK File
  - Example of, 24
  - Overview, 16
- INIARG.TLK
  - Description and Use, 25
- INIARG.TLK File, 120
  - Example of, 25
  - Overview, 16
- INPARG, 143
- INPINC, 144
- Input Flow Control
  - Overview, 18
- Interface
  - In Status Display, 29
- Internet Access
  - Accessing Netcom, 169
  - Establishing an Account, 168
  - Setting Up the Script, 168
  - Terminal Recommendations, 168
  - Using VersiCOMM, 167
- Internet file transfers, 194
- INUSE, 144

## J

- JWAIT, 144
- JWait
  - In Status Display, 29

## K

- KERMIT, 109

## L

- LABELCHR, 144
- LEXIS EMULATION, 125
- LINEFEED, 144
- Linefeed
  - In Status Display, 30
- LISTEN, 144
- LISTEN Command
  - Use in USING File, 67
- LOCAL, 144
- LOOKUP, 145
- ls
  - Additional information, 175
- lynx, 178
  - Adding a bookmark, 179
  - Additional Information, 181

- Commands, 180
- Defining Your Home Page, 180
- Exiting, 179
- Help, 179
- Hints, 179
- Starting, 178

## M

- MATCH, 41
  - Errors, 43
  - Operation, 41
  - Switches, 42
- MCI Mail
  - How to send MCI Letter, 210
- MCI QuikLetter
  - Overview, 18
- MINREC
  - To Receive Files from SEND, 72
- MINSND
  - To Send Files to RECEIVE, 93
- MLON and MLOFF, 41
- MODEM, 145
- Modem
  - Operational States, 201
  - Result Codes, 201
  - Talking to, 201
  - Troubleshooting, 202
- Modem Polling Files, 21
- Modem Setup Screen, 54
- Modem>Preventing Access To, 41
- MOVCUR, 146
- Multiple SAVE files, 26
- MVER, 146

## N

- Naming a Save File, 205
- Netcom
  - Command Aliasing, 172
  - Command recall, 172
  - Downloading Files Using utree, 177
  - Editing a File, 173
  - Electronic Mail, 186
  - File Management, 175
  - Getting help, 169
  - Gopher Server, 182
  - Logging OFF, 169
  - Newsgroups, 188
  - Other Sources of Information, 171
  - The login Shell, 169
- Netcom command
  - faq, 170

NEWLINE, 146  
Newsgroups, 188  
    Additional Information, 192  
    Finding, 190  
    Posting a Reply to a, 191  
    Reading an article in a, 189  
    Test newsgroup, 191  
    Using lynx to locate, 191  
    Using rn to locate, 190

NOTELOG, 146

NOTESAVE, 146

Nulls

    Embedded, 205  
    Filtering, 205

## O

OFF File

    Special Purpose USING File, 68  
    To Logoff Service, 68

Operation

    Changing to Command Mode, 27  
    COMM command format, 21  
    Idleout Exit, 27  
    Overview, 13  
    Preserve Screen Display, 27

Operational Switches

    In Status Display, 28

ORFLG, 146

Overview

    Command Mode, 14  
    Conversational Mode, 13

## P

PARAMCHR, 147

PARITY, 147

Parity

    In Status Display, 30

PAUSE, 147

PCHR, 148

PCMARG XCALL, 112

Port

    In Status Display, 30

Port Configuration

    In Status Display, 30

Port Selected

    In Status Display, 29

PRESERVE, 148

Preserve Screen

    Overview, 19

Preserving Displayed Data, 27

PRINT, 148

PROMPTCHR, 148

PTIME, 148

PWRITECR, 148

## Q

QUIT, 148

## R

RECEIVE, 149

Receiving Files

- Discussion, 92

- Using RECEIVE Command, 92

- Using RETRV, 93

- Using SAVE Command, 92

- XMODEM protocol, 104

- ZMODEM protocol, 106

RECORD, 149

RECX, 149

- Errors, 105

- Operation, 104

- Switches, 105

RECZ, 149

- Errors, 108

- Operation, 107

- Switches, 107

REDIAL, 151

REMOTE, 151

Remote System

- Changing Terminal Drivers, 44

RETRV, 151

- Errors, 101

- Operation, 94

- Switches, 97

- Troubleshooting, 100

RETRV Switches

- ADAPT, 97

- FASTRANS, 97

rn

- Additional Information, 192

- Getting help, 189

- Posting a Reply to a newsgroup, 191

- Reading a newsgroup, 189

RTS, 152

RTSLOW, 152

RUN, 152

RX

- Errors, 105

- Operation, 105

- Switches, 105

RZVA

- Use of FastQ, 75



## S

SAVE, 153

Save

    In Status Display, 28

Saving Displayed Information, 135, 138, 153, 205

    In Conversational Mode, 26

SELECT, 153

SEND, 153

    End-of-Line Delay, 73

    Flow Control, 73

    Linefeed Control, 73

    Speed Control, 72

    Throttle Parameter, 72

    Translating and Filtering, 73

SEND Parameters

    In Status Display, 30

SENDEPX, 154

    Specifying Packet Size, 74

Sending with X/ON X/OFF protocol, 142

SENDL, 154

SENDTLK, 155

SENDX, 155

    Errors, 89

    Operation, 88

    Switches, 89

SENDZ, 155

    Errors, 91

    Operation, 90

    Switches, 90

SETARG, 156

SETFLG, 156

SETPIN XCALL, 119

SHELL Menus

    Using with VersiCOMM, 39

SIZE, 157

Size

    In Status Display, 30

Soft Parity

    Overview, 19

SPARITY, 157

Special Features

    Overview, 14

STATUS, 157

Status Display

    Description, 28

    Example, 28

Status Line, 25

    Disabling, 25

    Overview, 19

STOP, 157

Stop

- In Status Display, 30
- STRIP, 158
  - Use When Receiving 8-bit Data, 92
- Strip
  - In Status Display, 29
- SUBFLG, 158
- SX
  - Errors, 89
  - Operation, 88
  - Switches, 89
- System Configuration
  - The INI.TIW File, 23

## T

- Talking to Your Own Computer, 204
- TALKTO, 158
- TALL, 158
- Task Manager
  - To Run VersiCOMM, 39
- telnet, 192
  - Additional Information, 194
  - Commands, 193
  - Example session, 193
  - Starting, 192
- Terminal
  - In Status Display, 29
- TERMINAL EMULATION, 123
- Terminal Emulation
  - Default, 123
  - Generic, 126
  - Keyboard Translation, 129
  - LEXIS, 125
  - Specifying, 124
  - Virtual, 123
  - VT100, 124
- Terminal Status
  - Checking Status Bits, 43
- The COMMCENTRE, 40
- TLOCAL, 158
- TLocal
  - In Status Display, 28
- TRANS, 158
  - Error Messages, 84
  - Operation, 76
  - Option Switches, 79
  - Troubleshooting, 83
- TRANS Operation
  - File List Processing, 78
- TRANS Switches
  - ADAPT, 79
  - FASTRANS, 79
  - FILELIST, 79

- MAXIMUM, 81
- PACKET, 81
- Transferring Files, 154, 155, 158
- Transferring Fils, 153
- Translating & Filtering Characters
  - Discussion, 38
- Translating Characters, 144, 151, 158, 159
  - Example of, 205
  - Overview, 17
- Transmitting Files
  - Discussion, 71
  - Using SEND Command, 72
  - Using SENDEPX Command, 73
  - Using SENDL, 74
  - Using SENDTLK, 74
  - Using TRANS, 75
  - With Echo Error Detection, 73
  - XMODEM protocol, 87
  - ZMODEM protocol, 89
- TREAD
  - Examples, 116
  - Using, 113
- TREAD XCALL, 112
- TREMOTE, 159
- TRemote
  - In Status Display, 29
- trmdef.HU0
  - Special Purpose USING File, 68
  - To Hangup Smart Modem, 68
- TRMFLG, 159
- TRMSTS, 43
- Troubleshooting
  - Connecting to another computer, 203
  - Talking to Modem, 202
- TSTARG, 160
- TSTDIALOG, 160
- TSTFLG, 160
- TWAIT
  - Examples, 117
- TWAIT XCALL, 116
- TWRITE
  - Examples, 113
  - Using, 113
- TWRITE XCALL, 112
- TYPE, 161
- TYPECR, 161
- TYPEFLG, 161
- TYPEWAIT, 161

## U

- Unix
  - Command Aliasing, 172

- Command recall, 172
- Unix command
  - alias, 172
  - apropos, 170
  - Common commands, 170
  - elm, 187
  - ftp, 194
  - history, 172
  - ls, 175
  - lynx, 178
  - man, 169
  - pico, 173
  - rn, 189
  - telnet, 192
  - utree, 176
- Unix commands
  - Getting help, 169
- USING File
  - Command and Comment Characters, 62
  - Description and Use, 61
  - Format, 62
  - HANGUP.TLK File, 68
  - How to write, 206
  - OFF Files, 68
  - Overview, 17
  - Termination, 68
  - To Access Compuserve, 206
  - To Access EasyLink, 62
  - Use of CCENDTLK Command, 66
  - Use of CTRLC Command, 65
  - Use of DIALOG Command, 66
  - Use of dollar-sign, 62
  - Use of LISTEN Command, 67
  - Use of semi-colon, 62
  - Use of WAIT Command, 64
  - Use of WASIT Command, 65
- Using RUN on 2.x Systems, 111

## V

- VersiCOMM Commands
  - Entering, 31
- VersiCOMM Initialization
  - The INI.TLK File, 24
- VersiCOMMander
  - Files Submenu, 34
  - Key commands, 33
  - Main Window, 33
  - Receive Submenu, 36
  - Selecting files, 33
  - Send Submenu, 36
  - Settings Submenu, 37
  - Starting, 32

- Status Window, 34
- System Submenu, 38
- VIRTUAL TERMINAL EMULATION, 123
- VPSTRN, 129
- VT00 Emulation
  - Overview, 18
- VT100 EMULATION, 124

## W

- WAIT, 161
- WAIT Command
  - Conditional Branching, 64
  - Use in USING File, 64
- WAITIME, 162
- WASIT, 162
- WASIT Command
  - Use in USING File, 65
- World Wide Web
  - Browsing, 177
  - California Department of Water Resources Snow Page, 177
  - FedEx Airbill Tracking Form, 178
  - Internet Assistance and Information, 179
  - lynx, 178
  - Prime Minister of Japan official residence, 178
  - Yahoo and Other Search Utilities, 180
- WPAUSE, 163
- WPAUSECHR, 163
- WPREFIX, 163
- WPREFIXCHR, 163
- WRAP, 163
  - And gopher, 182
- WRITE, 163
- WRITECR, 164

## X

- X command, 164
- XON/XOFF
  - HANDSHAKE Switch, 73
  - Using with SEND, 73
- XORFLG, 164

## Y

- Yahoo, 180

## Z

- ZAUTO, 165
- ZDLE, 165