

**Revision 3.1
March 19, 1999**

AccuTerm 97

User Guide

**AccuSoft
8041 Foothill Blvd.
Sunland, California 91040**

**www.asent.com
Phone (818) 951-1891
FAX (818) 951-3606**

AccuTerm and AccuPlot are trademarks of Schellenbach & Associates, Inc.

Microsoft and Microsoft Windows are registered trademarks of Microsoft Corp.

IBM is a registered trademark of International Business Machines Corp.

Tektronix 4010/4014/4100 is a trademark of Tektronix, Inc.

Viewpoint is a trademark of Applied Digital Data Systems.

Wyse-50 and Wyse-60 are trademarks of Wyse Technology.

DEC, VT-220 and VT-52 are trademarks of Digital Equipment Corp.

Pick is a trademark of Pick Systems.

PicLan is a trademark of Modular Software Corporation.

Procomm is a trademark of DataStorm Technologies, Inc.

Intel is a trademark of Intel Corporation.

Hayse is a trademark of Hayse Microcomputer Products.

Software License Agreement

This AccuSoft software product ("Product") is a proprietary and copyrighted Product of Schellenbach & Associates, Inc. dba AccuSoft Enterprises ("ASE"). It is licensed (not sold) to you, and you may use this Product only on the terms of this license from ASE, even though you may have obtained the Product from a dealer or other source.

License

ASE grants you a non-exclusive license only to:

1. use the Product on one computer which you own or use. A Separate license is required for use on each other Computer on which the Product used. But, you may transfer the product to another computer if it is a replacement for the original computer or if the original computer is inoperable due to a malfunction, until it is repaired.
2. merge the Product with other applications which utilize the Product. This license will remain effective as to any merged portion of this Product.
3. make one copy of the Product for archival or backup purposes in support of your use of the Product.
4. transfer the Product and this license to another party if the other party agrees to accept the terms and conditions of this Agreement. If you transfer the Product, you must at the same time either transfer or destroy all copies, including any portions of the product merged into other programs or stored on computer memory devices such as hard disks. IF YOU TRANSFER POSSESSION OF ANY COPY OR MERGED PORTION OF THE PRODUCT TO ANOTHER PARTY WITHOUT TRANSFERRING THE PRODUCT ITSELF AS PROVIDED HERE, YOUR LICENSE IS AUTOMATICALLY TERMINATED.

Except as otherwise provided in this agreement, you have no right under this license to rent, lease, transfer, or distribute the Product. You may use the Product in object code form only.

Title

Title and copyright to the Product, including the Product media and documentation, remain with ASE. You are not the owner of this Product, but you have the right to use it in accordance with this license. You are not licensed to copy, reproduce or make data transmissions, in whole or in part, of the Product, except as is permitted in this license. You agree not to modify, reverse engineer, translate, disassemble or decompile the Product, in whole or in part.

License Term

The license is effective upon opening of the Product package or use of the Product and shall continue until terminated. You may terminate it at any time by destroying the Product together with all copies and the Product Documentation. THIS LICENSE WILL TERMINATE AUTOMATICALLY IF YOU FAIL TO COMPLY WITH ANY TERM OR CONDITION OF THIS LICENSE. Upon termination you agree to destroy the Product, the Product Documentation and all copies, purge the Product from all computers and memory devices such as hard disks, and eliminate the Product from any merged programs.

Limited Warranty

ASE warrants that for a period of six(6) months from date of purchase the Product will perform according to the specifications set forth in the associated Product Documentation and the diskette (or other medium) on which the Product is located will be free from defects in materials and workmanship under normal use.

YOU ASSUME THE RESPONSIBILITY FOR THE SELECTION OF THE PRODUCT TO ACHIEVE YOUR INTENDED RESULTS, AND FOR THE INSTALLATION, USE AND RESULTS OBTAINED FROM THE PRODUCT. ASE does not warrant that the

functions contained in the Product will meet your requirements or that the operations of the Product will be uninterrupted or error free, or will function properly in every hardware environment. This warranty gives you specific legal rights and you may also have other rights which vary from state to state.

EXCEPT AS EXPRESSED HEREIN, THE LICENSED PROGRAM IS PROVIDED TO YOU "AS IS". ASE DISCLAIMS TO THE FULLEST EXTENT PERMITTED BY LAW ALL OTHER WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, WHICH SHALL IN NO EVENT EXCEED ONE YEAR. Some states do not allow the exclusion of implied warranties so the above exclusion may not apply to you.

No person is authorized to modify the warranties contained in this license or make any different warranty for the Product.

Limitations of Remedies

ASE's entire liability and your exclusive remedy shall be limited to:

1. the replacement of any Product, including its diskette or other medium, that does not meet the standards of the limited warranty above. To use this warranty, you must return, at your expense, the Product and all copies along with a copy of your purchase invoice and a brief description of the defect to AccuSoft Enterprises, Customer Service Dept., 8041 Foothill Blvd., Sunland, CA 91040. Upon verification of the defect, ASE will send to you, at ASE's expense, a replacement Product.
2. If ASE is unable to deliver a replacement which is free of defects, ASE will refund your purchase price.

IN NO EVENT WILL ASE BE LIABLE TO YOU FOR ANY DAMAGES, INCLUDING ANY LOST PROFITS, LOST SAVINGS, LOST DATA OR OTHER INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PRODUCT EVEN IF ASE OR ANY OF ITS DEALERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES, OR FOR ANY CLAIM BY ANY OTHER PARTY. ASE'S MAXIMUM LIABILITY SHALL IN NO EVENT EXCEED THE PURCHASE PRICE YOU PAID FOR THE PRODUCT. Some states do not allow the limitation or exclusion of liability for incidental or consequential damages so the above limitation or exclusion may not apply to you.

Customer Registration

ASE may from time to time revise and update its Product or provide useful customer information. You may wish to know about Product improvements and the availability of upgrades. To obtain this useful service, you must fill out and return the enclosed customer registration card. ASE urges you to do so immediately.

General

You may not sublicense, assign or transfer this license or the Product to any other person or entity except as expressly allowed in this license. Any attempt to sublicense, assign or transfer is void.

This Agreement will be governed by laws of the State of California, exclusive of its choice of law provisions.

You further agree that this is the complete and exclusive statement of the agreement between us which supersedes any proposal or prior agreement, oral or written, and any other communications between us relating to the subject matter of this agreement.

If any provision of this agreement shall be held to be invalid, illegal or unenforceable, the remaining provisions shall not be affected.

If you have any questions concerning this license, you should contact AccuSoft Enterprises, Customer Service Dept., 8041 Foothill Blvd., Sunland, CA 91040.

Software Site License Agreement

This AccuSoft software product ("Product") is a proprietary and copyrighted Product of Schellenbach & Associates, Inc. dba AccuSoft Enterprises ("ASE"). It is licensed (not sold) to you, and you may use this Product only on the terms of this license from ASE, even though you may have obtained the Product from a dealer or other source.

License

ASE grants you a non-exclusive license only to:

1. use the Product on one or more computers you own or use at a single location or facility, solely for use by persons within that location or facility. You may make copies of this Product as necessary to use the Product with other Computers at your licensed location or facility. A separate license is required for each location or facility at which the Product is used.
2. merge the Product with other applications which utilize the Product. This license will remain effective as to any merged portion of this Product.
3. make one copy of the Product for archival or backup purposes in support of your use of the Product.
4. transfer the Product and this license to another party if the other party agrees to accept the terms and conditions of this Agreement. If you transfer the Product, you must at the same time either transfer or destroy all copies, including any portions of the product merged into other programs or stored on computer memory devices such as hard disks. IF YOU TRANSFER POSSESSION OF ANY COPY OR MERGED PORTION OF THE PRODUCT TO ANOTHER PARTY WITHOUT TRANSFERRING THE PRODUCT ITSELF AS PROVIDED HERE, YOUR LICENSE IS AUTOMATICALLY TERMINATED.

Except as otherwise provided in this agreement, you have no right under this license to rent, lease, transfer, or distribute the Product. You may use the Product in object code form only.

Title

Title and copyright to the Product, including the Product media and documentation, remain with ASE. You are not the owner of this Product, but you have the right to use it in accordance with this license. You are not licensed to copy, reproduce or make data transmissions, in whole or in part, of the Product, except as is permitted in this license. You agree not to modify, reverse engineer, translate, disassemble or decompile the Product, in whole or in part.

License Term

The license is effective upon opening of the Product package or use of the Product and shall continue until terminated. You may terminate it at any time by destroying the Product together with all copies and the Product Documentation. THIS LICENSE WILL TERMINATE AUTOMATICALLY IF YOU FAIL TO COMPLY WITH ANY TERM OR CONDITION OF THIS LICENSE. Upon termination you agree to destroy the Product, the Product Documentation and all copies, purge the Product from all computers and memory devices such as hard disks, and eliminate the Product from any merged programs.

Limited Warranty

ASE warrants that for a period of six(6) months from date of purchase the Product will perform according to the specifications set forth in the associated Product Documentation and the diskette (or other medium) on which the Product is located will be free from defects in materials and workmanship under normal use.

YOU ASSUME THE RESPONSIBILITY FOR THE SELECTION OF THE PRODUCT TO ACHIEVE YOUR INTENDED RESULTS, AND FOR THE INSTALLATION, USE AND RESULTS OBTAINED FROM THE PRODUCT. ASE does not warrant that the

functions contained in the Product will meet your requirements or that the operations of the Product will be uninterrupted or error free, or will function properly in every hardware environment. This warranty gives you specific legal rights and you may also have other rights which vary from state to state.

EXCEPT AS EXPRESSED HEREIN, THE LICENSED PROGRAM IS PROVIDED TO YOU "AS IS". ASE DISCLAIMS TO THE FULLEST EXTENT PERMITTED BY LAW ALL OTHER WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, WHICH SHALL IN NO EVENT EXCEED ONE YEAR. Some states do not allow the exclusion of implied warranties so the above exclusion may not apply to you.

No person is authorized to modify the warranties contained in this license or make any different warranty for the Product.

Limitations of Remedies

ASE's entire liability and your exclusive remedy shall be limited to:

1. the replacement of any Product, including its diskette or other medium, that does not meet the standards of the limited warranty above. To use this warranty, you must return, at your expense, the Product and all copies along with a copy of your purchase invoice and a brief description of the defect to AccuSoft Enterprises, Customer Service Dept., 8041 Foothill Blvd., Sunland, CA 91040. Upon verification of the defect, ASE will send to you, at ASE's expense, a replacement Product.
2. If ASE is unable to deliver a replacement which is free of defects, ASE will refund your purchase price.

IN NO EVENT WILL ASE BE LIABLE TO YOU FOR ANY DAMAGES, INCLUDING ANY LOST PROFITS, LOST SAVINGS, LOST DATA OR OTHER INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PRODUCT EVEN IF ASE OR ANY OF ITS DEALERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES, OR FOR ANY CLAIM BY ANY OTHER PARTY. ASE'S MAXIMUM LIABILITY SHALL IN NO EVENT EXCEED THE PURCHASE PRICE YOU PAID FOR THE PRODUCT. Some states do not allow the limitation or exclusion of liability for incidental or consequential damages so the above limitation or exclusion may not apply to you.

Customer Registration

ASE may from time to time revise and update its Product or provide useful customer information. You may wish to know about Product improvements and the availability of upgrades. To obtain this useful service, you must fill out and return the enclosed customer registration card. ASE urges you to do so immediately.

General

You may not sublicense, assign or transfer this license or the Product to any other person or entity except as expressly allowed in this license. Any attempt to sublicense, assign or transfer is void.

This Agreement will be governed by laws of the State of California, exclusive of its choice of law provisions.

You further agree that this is the complete and exclusive statement of the agreement between us which supersedes any proposal or prior agreement, oral or written, and any other communications between us relating to the subject matter of this agreement.

If any provision of this agreement shall be held to be invalid, illegal or unenforceable, the remaining provisions shall not be affected.

If you have any questions concerning this license, you should contact AccuSoft Enterprises, Customer Service Dept., 8041 Foothill Blvd., Sunland, CA 91040.

Contents

Introduction	1
Features of AccuTerm 97	1
Hardware and Software Requirements	3
Getting Started.....	5
Installation.....	5
Network Installation	7
Running AccuTerm 97	9
AccuTerm Main Menu	12
Configuring AccuTerm.....	19
Global Settings	19
General Settings	20
File Transfer Settings	22
Session Settings.....	25
Device Settings.....	25
Terminal Type Settings	38
Screen Settings	40
Font Settings.....	45
Color Settings.....	47
Keyboard Settings	49
Printer Settings	51
Miscellaneous Settings	54
Auto Dialer	57
File Transfer	61
Data Capture.....	70
Scripting	72
AccuTerm Object Reference	85
AccuTerm Programming	137
Wyse Programming	147
ADDS Programming	164
ANSI Programming.....	171
Pick PC Console Programming	187
Appendix A - Wyse Tables	193
Appendix B - Viewpoint Tables	199
Appendix C - ANSI Tables	204
Appendix D - ASCII Codes.....	208
Appendix E - Pick File Transfer and Server.....	212
Appendix F - Object Bridge	242
Appendix G - Custom Features	248

INTRODUCTION

Welcome to **AccuTerm 97**. We've worked hard to give you the fastest and most accurate terminal emulation package on the market. We have also tried to make it easy to use. Our philosophy has been to add the features that will be most useful without sacrificing ease of use, accuracy or performance.

Features of AccuTerm 97

Multiple sessions using **MDI** (Multiple Document Interface). This means that you can run two or more terminal sessions at the same time without running two copies of AccuTerm. It works the same as other programs that allow you to work with more than one document at a time like word processors, spreadsheets, etc. so it's easy to work with. You don't have to learn a new way of working. Also, by using MDI instead of running multiple copies of AccuTerm, you use less system resources and memory. You can still run multiple copies if you choose, we just think MDI works better.

Full 32 bit application designed for Windows 95, 98 and NT.

User interface has Office 97 look and feel, including context menus accessed by clicking with the right mouse button. The new toolbar can be hidden completely and also features balloon help that shows you what the buttons mean. Tool bar, menu and status bar can be customized by the user.

Full OLE Automation / ActiveX support allows integration with other Windows applications. AccuTerm 97 acts as both automation client and automation server. AccuTerm exposes its application and session objects, which allows other applications complete control of AccuTerm.

ActiveX component (OCX) terminal emulation & communications engine may be used in custom VB applications where accurate terminal emulation is required.

Accurate emulations. When you use AccuTerm everything will work just like the terminal being emulated. No strange problems with word processors or spreadsheets. No messed up screens. No hassles.

Modem connections use Windows built-in dialer, and unlimited phone directory entries.

File transfer capabilities using industry standard protocols like Xmodem, Ymodem, Zmodem, and Kermit. Pick file transfer utilities include text and data files and direct Pick-to-Pick file transfer.

Scripting language allows users to write scripts and macros using Visual Basic for Applications

Enhanced scroll back window saves up to 1200 lines for instant recall.

Extended screen support of up to 160 columns by 60 rows.

Automatic font scaling so that you can always see the full screen. This is a great feature that once you get used to it, you'll never want to go without.

Enhanced network support for PicLan and Telnet (TCP/IP).

Tabbed setup screens efficiently organize settings and preferences.

Image and sound support lets you to display images anywhere on the terminal screen. You can also play Wave and MIDI sound files.

Tektronix graphics terminal emulation in both color and monochrome.

Full color mapping lets you choose any foreground and background color combination for any terminal visual attribute.

Easy to use. We've followed the Windows Interface Application Design Guide more closely than many other terminal emulation programs. This makes AccuTerm easier to learn and use.

Hardware and Software Requirements

Memory:	16MB
Processor:	Pentium 90 or better
Windows:	95, 98, NT 3.51, NT 4.0
COM port:	Any serial port or internal modem supported by Windows
Mouse:	Any mouse supported by Windows

GETTING STARTED

This manual assumes that you have a working knowledge of the Windows environment. If you are new to Windows, there are many good books available. They do a much better job of explaining the basics than we could do in this manual.

Installation

To install AccuTerm 97, run the SETUP program from the AccuTerm 97 diskette # 1 or CD.

- 1) Start the setup program:

Diskette installation: click the **Start** button and select **Run** from the start menu. Complete the **Run** dialog box with **a:setup**

CD installation: insert the CD and select “Install AccuTerm 97” from the menu. If the setup program does not automatically start when the CD is inserted, then follow the *diskette installation* instructions above, but change the drive letter to the letter for your CD drive.

- 2) The AccuTerm 97 installation wizard will guide through the installation process. You will be prompted to enter registration information, select the destination directory and select a start menu program folder. You move forward through the installation process by clicking the **Next** button. You can click the **Back** button to move back, or the **Cancel** button to cancel the installation.
- 3) The AccuTerm Registration dialog box will be displayed. If this is the first time AccuTerm has been installed, you need to fill in the user, company and location in this dialog, then click the **Next** button. If AccuTerm has already been installed, the

licensed user, company and location will be displayed; simply press click the **N**ext button.

- 4) Select the directory where you want the AccuTerm program files to be installed, then click the **N**ext button.
- 5) If the drive you are installing AccuTerm to is a “network” drive, you will need to select a “home” directory. Select a local or private, non-shared directory where AccuTerm can store settings and configuration files for your computer. After selecting the home directory, click the **N**ext button.

Note: if the drive you are installing AccuTerm to is a “network” drive, but the directory is not shared by other network users, AccuTerm can use the program directory for saving settings.

- 6) You can choose if you want files which are replaced during installation to be saved in a backup directory. If you select this option, when you click the **N**ext button, you can select the directory where replaced files are saved to.
- 7) Select a “Start Menu” folder for the AccuTerm 97 shortcut, then click the **N**ext button. You can also place a shortcut to AccuTerm 97 on your desktop.
- 8) AccuTerm 97’s standard imaging feature supports the most commonly used image formats. If you require support for other image formats, select “Include optional image format support”, then click the **N**ext button.

Note: if you do not require the optional image formats, do not check this option to conserve resources when running AccuTerm 97.

- 10) To begin installing files, click the **N**ext button. The installation progress will be displayed. When the files from the first diskette have been installed, you will be prompted to insert the second diskette, then click the **O**K button. Repeat for each remaining diskette. When the installation is complete, you can view the release notes by checking the “View release notes” box, and clicking **F**inish. If you do not want to view the

release notes at this time, simply clear the “View release notes” box and click **Finish**.

During installation, you may get a message indicating that the OLE system files are in use. If this occurs, please shut down all other applications, then click the **OK** button to restart Windows. After Windows restarts, re-start the installation procedure.

Network Installation

Installing from a network drive

Diskettes

If you have a network and are installing a site license version of AccuTerm from diskettes, you can copy the contents of the distribution disks to a shared directory on the network, for example “ATWDISKS”. ***Be sure to install AccuTerm from the diskettes on one workstation before copying the diskette contents to the shared network directory, as certain files may not get copied to the shared directory until AccuTerm has been installed from the diskettes at least one time.***

CD-Rom

If you have a network and are installing a site license version of AccuTerm from CD, you can select the “Copy setup files to server” option from the startup screen displayed when the CD is inserted. You will need to enter your registration information and CD Key at this time.

After copying the setup files to the shared directory, go to each workstation on which you want to install AccuTerm, and run SETUP.EXE from the shared directory: In step 1 above, change **a : setup** to **d : \atwdisks\setup** (change **d** to your network drive letter).

Even if you are installing AccuTerm on a network, you still *must install it on each workstation*. This is because some files are installed in non-shared directories such as WINDOWS and WINDOWS\SYSTEM.

Sharing AccuTerm on a network

If you want to install a single, shared copy of AccuTerm, you will need a local or private, non-shared directory where AccuTerm can store its settings and configuration files. Specify this directory in step 5 above.

Directories

AccuTerm saves general settings in the file **ATWIN.INI**. This file is normally written to the same directory as the program (or the “home” directory specified in step 5 above). This is also the default directory where AccuTerm looks for configuration files. You can change this directory by setting an environment variable:



SET ATWIN=d: *path*

Note that there is no space on either side of the equal sign. Replace **d** and ***path*** with the drive and directory where you want to save the ATWIN.INI file. Place the **SET** command in your AUTOEXEC.BAT file.

You can also specify the directory where AccuTerm saves the ATWIN.INI file by adding an [ATWIN] section to the WIN.INI file. Insert the item “HomePath=d: ***path***” in the [ATWIN] section. Replace **d** and ***path*** with the drive and directory where you want to save the ATWIN.INI file. When the installation program detects that AccuTerm is being installed on a network, it updates the WIN.INI file with the name of the private home directory.

Running AccuTerm 97


AccuTerm 97 uses the Windows MDI (Multiple Document Interface) standard for handling more than one session. It works the same as other popular Windows applications like Word 97, Excel 97, Lotus 123, except that instead of opening documents, you open terminal configuration files. Each configuration file that you open becomes one session.

To run AccuTerm 97, simply select  AccuTerm 97 from the Start menu, or click the AccuTerm icon  on the desktop. AccuTerm's main window will be displayed:





At the start, AccuTerm shows a blank screen with only the **File**, **View**, **Tools** and **Help** menu options available. This is because no sessions are open.

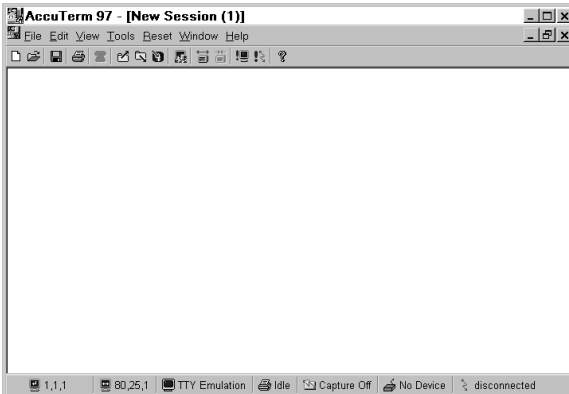
Creating a New Session

After starting AccuTerm, you can use the **Connection Wizard** to guide you through the process of creating a new session. To start the Connection Wizard, select **N**ew from the **F**ile menu or click the  button on the toolbar.



Click the  button to start the wizard, then follow the step by step instructions for creating a new session. If you do not wish to use the wizard, simply click the  button.

After opening a new session the screen will look like this:



Once a session has been opened, the menu changes to **F**ile, **E**dit, **V**iew, **T**ools, **R**eset, **W**indow, and **H**elp. The status bar is also displayed at the bottom of the main window.

Command Line

The name of a saved configuration (.CFG), layout (.LYT) or script (.SCR) file may be appended to the end of the ATWIN command line. When a file name is specified in the command line, AccuTerm will open that file upon

start up. In this manner, AccuTerm can open a single session, an entire layout, or run a script when the program shortcut is double-clicked.

To modify the command line, click the Windows **Start** button with the *right* mouse button, then select **E**xplore. Locate the AccuTerm shortcut, then click the shortcut entry with the *right* mouse button, and select **P**roperties. Click on the Shortcut tab, and click in the Target box. Press the **END** key, then type a space and the name of the CFG, LYT or SCR file. Click the **OK** button to save changes. Close the Exploring Start Menu window.


Note: to load a layout and run a script, open the layout file, load the script, then save the layout. Append the layout file name to the command line. When opening the layout file, first all session files contained in the layout will be opened, the script's Main subroutine will be run.

AccuTerm Main Menu


File Menu

The AccuTerm **F**ile menu contains options for creating sessions, opening and closing sessions, and saving sessions as well as printer setup. The most common menu options may also be executed by clicking a button on the toolbar.

File New

To create a new session, select the **N**ew option or click the  button on the toolbar. The New Connection panel will be displayed. Select **C**onnection **W**izard to use the connection wizard to help you build a new session, or select **N**ew **C**onnection to create a new session.


File Open

To open an existing configuration file, select the **O**pen option or click the  button on the toolbar. This creates a new session with the settings saved in the selected configuration file.

File Close

To close the active session, select the **C**lose option. If you have made any changes to the session settings, you will have the opportunity to save them in a configuration file before closing the session.

File Save and File Save As

After adjusting session settings, save the new settings in a configuration file by selecting the **S**ave or **S**ave **A**s option, or by clicking the  button.


File Save Layout and File Save Layout As

If you have several sessions open, you can save them as a group, or *layout*, by selecting the **S**ave **L**ayout or **S**ave **L**ayout **A**s options.

File Printer On

Selecting this option toggles the printer port on and off. When the printer is on, received data is printed on the selected printer and displayed on the terminal screen. You can also press **ALT + P** to toggle printer the printer port status.

File Print Screen

Selecting this option or clicking the  button causes the current terminal screen to be printed on the selected printer. You can also press **ALT + G** to print the terminal screen.

File Print Setup

Select this option to set up printer information. See the section on “Printer Selection” in the next chapter for details.

Recent file 1 to 4

Opens one of the last four most recently opened configuration or layout files.

File Exit

Select this option to terminate AccuTerm.

Edit Menu

The AccuTerm **Edit** menu contains commands to copy and paste from the Windows clipboard.

Edit Copy

This command will copy a selected block of text from the terminal screen to the Windows clipboard. If there is no selection, the entire screen is copied. To select a block on the screen, position the mouse on the upper left corner of the desired block. Press and hold the left mouse button for about one half second, then drag the mouse to the lower right corner, and release the button. The selected area will be indicated by a dotted rectangle. To remove the selection, click the left mouse button anywhere on the screen. If the screen needs to be scrolled to select the desired data, move the mouse to the edge of the window and the window will automatically scroll in that direction.

Edit Copy History

This command will copy the contents of the history buffer to the Windows clipboard.

Edit Paste

This command will send the clipboard contents to the host computer. Each line is terminated as specified in the ASCII file transfer settings (see **File Transfer Settings**).



Edit Paste From

This command will send the contents of a text file to the host computer. A file selection dialog box will be opened when you select this command.



View Menu

The AccuTerm **View** menu contains options to select screen size, lock/unlock history, and display or hide the toolbar and status line.

View Normal Screen Size and View Extended Screen Size

Changes the number of columns and rows on the terminal screen to the Normal size (usually 80 columns) or Extended size (usually 132 columns). See Screen Settings for information on setting the number of columns and rows. The screen size may also be changed by clicking the  or  buttons.

View Lock History and View Unlock History

Locks or unlocks updates to the history buffer. Locking the history buffer prevents lines which scroll off of the terminal screen from being saved in history. The history buffer may also be locked or unlocked by clicking the  or  buttons.

View Status Line

Check this option to display the active session status at the bottom of AccuTerm's main window. Un-check this option if you do not want the status displayed.


View Toolbar

Check this option if you want the toolbar displayed in AccuTerm's main window. Un-check this option if you want the toolbar hidden.


Tools Menu

The **Tools** menu contains options for modifying AccuTerm's settings, dialing a number, initiating file transfers and data capture, and opening the script window. Each of these options opens a dialog box where the desired function is performed. See the appropriate section in this manual for details on each function.


Tools Settings

Select this option or click the  button to open the Settings dialog box. This dialog is used to modify all of AccuTerm's settings and preferences, such as communications device, terminal type, screen colors, etc.


Tools Auto Dial

Select the **Auto Dial** option to enter or choose a phone number to dial when using a Modem connection. Also choose this option to disconnect a Modem connection. Note: this option will be disabled unless the connection device type is "Modem". You can quickly call up the Auto Dial dialog by clicking the  button on the toolbar.


Tools Upload

Selecting this option displays the AccuTerm File Upload dialog box. Use this dialog to select files to upload, select a transfer protocol, initiate an upload and specify other options. You can quickly call up the Upload dialog by clicking the  button on the toolbar.

Tools Download

Selecting this option displays the AccuTerm File Download dialog box. Use this dialog to select the destination download directory, select a transfer protocol, initiate a download and specify other options. You can quickly call up the Download dialog by clicking the  button on the toolbar.

Tools Capture

Selecting this option displays the Begin Capture or End Capture dialog box. Use the Begin Capture dialog to select the destination file and specify other options. Use the End Capture dialog to terminate the capture operation. You can quickly call up the Capture dialog by clicking the  button on the toolbar.

Tools Script

This option opens the AccuTerm Script Window, where you can load script files and edit and run macros.

Reset Menu

This menu has options to reset the terminal or the connection.

Reset Terminal

Resets the terminal (clears screen, restores default character set and emulation, etc. You can quickly reset the terminal by clicking the



button on the toolbar.

Reset Connection

Resets the connection. If the session has been disconnected, this command will attempt to reconnect. You can quickly reset the connection by clicking the



button on the toolbar.

Window Menu

The Window menu contains options for re-arranging the open sessions within the main window, and for switching from one session to another.

Window Cascade

This option arranges all open session windows so that they overlap. The session title of all sessions remains visible.

Window Tile Horizontally and Window Tile Vertically

These options arrange all open session windows so that no windows overlap.

Windows

The open sessions are shown at the end of the **Window** menu. You can switch to any session by clicking on its entry in this menu.

Help Menu

You can get context sensitive help at any time by pulling down the **Help** menu and selecting the **Index** or **Contents** option or by clicking the



button. To display information about AccuTerm, including version, release, serial number and licensed user, select the **About AccuTerm 97** option.

Other Keyboard Commands

Toggle Status Line - ALT + L

This command toggles the *terminal's* status line, displayed as the last line on the terminal screen, between function key labels, host message and none. This command does not affect the AccuTerm status bar shown at the bottom of the session window. Use the **View** menu to show or hide the AccuTerm status bar.

Print Screen - ALT + G

Prints the contents of the terminal screen to the printer using the currently selected printer. The screen is printed in text mode (not graphic mode).

Data Capture - ALT + C

This is a shortcut to open up the Capture dialog box.

DOS Prompt - ALT + D

Opens a DOS session by running COMMAND.COM. Type `exit` to terminate the DOS session.

Toggle Printer Port- ALT + P

Toggle printer port on and off.

Previous Page - CTRL + PGUP

Displays the previous terminal page. To use this command, the session must be setup with more than one video page.

Next Page - CTRL + PGDN

Displays the next terminal page.

Pause / Resume - SCROLL LOCK


This command pauses or resumes the processing of received data by the terminal. The keyboard "Scroll Lock" LED indicates the current status.

Scanning the History Buffer

Use **CTRL + SHIFT + PGUP**, **CTRL + SHIFT + PGDN**, **CTRL + SHIFT + UP**, **CTRL + SHIFT + DOWN**, **CTRL + SHIFT + HOME**, and **CTRL + SHIFT + END** to navigate the history buffer.

CONFIGURING ACCUTERM

Before you can use AccuTerm, you must configure the program. The configuration process is used to tailor AccuTerm to your specific environment: workstation, printers, network and host computer. You can also choose certain preferences. This section describes the configuration process.

To modify AccuTerm's settings choose the **S**ettings item from the **T**ools menu or click the  button on the toolbar.

The **S**ettings dialog box is organized into ten categories, with each category on a separate index tab. Click on one of the index tabs to display the settings for that category. You can also press the number of the category while holding the **ALT** key (use **ALT** + 0 for 10 – Misc).

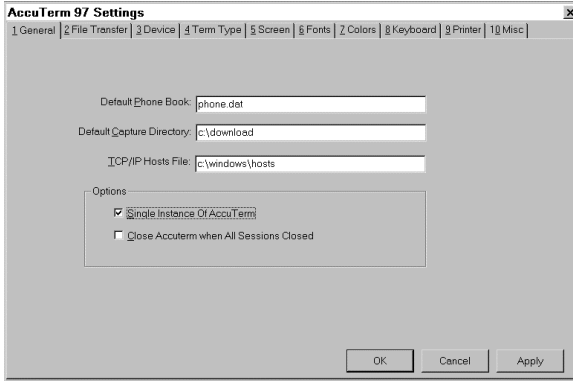
When you have completed your changes, simply press **ENTER** (or click the **OK** button) to save changes, or press **ESC** (or click the **Cancel** button) to discard changes and close the **S**ettings dialog box.

Global Settings

AccuTerm has two kinds of settings: *global*, which affect all sessions, and *session*, which affect only a specific session. Global settings should be checked after installation and when your operating environment changes.

The first two categories in the **S**ettings dialog are used to change *global* settings. These categories are **General**, where certain file names and other preferences are specified, and **File Transfer**, where file transfer protocol options may be modified.

General Settings



Default phone book

This field contains the file name to be used as the default phone book for the Auto Dialer. If no directory is specified, the current directory is assumed. The phone book file is a tab delimited ASCII file, with one line per entry; name and phone number are the two fields.

Default capture directory

This field contains the default directory for storing captured data. See the “Data Capture” section of this manual for details on this feature.

TCP/IP hosts file

When setting up a session to use Telnet over a TCP/IP network, the hosts file is used to present a list of valid host names for the user to choose from. If this file is not specified, the HOSTS file in your Windows directory will be used if it exists. Note: if this file is not specified or cannot be accessed, you may need to enter the IP address of the desired host in order to set up a Telnet session.

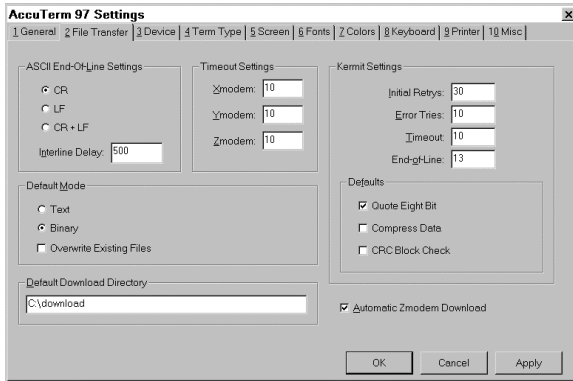
Single instance of AccuTerm

If this option is checked, only one instance of the AccuTerm program is allowed to be running. If you try to start AccuTerm from the Start menu when AccuTerm is already running, the running program will be activated. When this option is not checked, a new instance of the AccuTerm program will be started; that is you will have two or more AccuTerm tasks active.

Close AccuTerm when all sessions closed

If this option is checked, when the last open session is closed, the AccuTerm program will automatically close. This option is useful for network users when opening a configuration or layout file from the command line so that when all network sessions have been disconnected, AccuTerm will close automatically, without user intervention.

File Transfer Settings



This section is used to set up file transfer protocol options. There are sections for ASCII options, X/Y/Zmodem options, Kermit options, as well as options for default modes and directories. The default values for these options are generally OK, but you may need to alter some of them for reliable file transfers in your environment.

ASCII settings

When transferring a file from the PC to a host computer (upload) using the ASCII protocol, the each line may be terminated with a **CR**, **LF** or **CR LF** combination. Select the appropriate end of line for the receiving machine by checking the appropriate radio button.

ASCII is a “blind” protocol. The receiving machine cannot inform the sending machine if it is ready to receive. In order to perform reliable ASCII uploads, you may need to specify an Interline delay time. After each line is sent to the receiving machine, AccuTerm will wait for the specified delay time before sending the next line. Specify this value in milliseconds.

These settings also apply to the **Edit Paste** and **Edit Paste From** commands.

Timeout settings

When using any of the X/Y/Zmodem protocols, you can specify the number of seconds before a timeout occurs. If the other machine fails to respond within the number of seconds specified here, the protocol will attempt to resend the previously sent message.

Kermit settings

The Kermit file transfer protocol is popular with minicomputer and mainframe systems. There have been many revisions of this protocol, and it may be necessary to modify some of these settings to communicate with another Kermit program.

Kermit settings: Defaults

These default options may be set (checked) or reset (un-checked) as desired. When AccuTerm’s Kermit is negotiating with the other Kermit program, any of the checked options will be requested, and if the other Kermit is capable, the option will be used.

The “Quote eighth bit” option should be used when the communications path only supports 7 data bits, or when the other machine is not capable of processing eight bit data.

The “Compress data” option may be used to perform rudimentary data compression. Not all Kermit programs perform data compression.

The “CRC block check” option instructs Kermit to use CRC error detection instead of the standard simple checksum error check. This results in more reliable transfers, especially over noisy phone lines. Not all Kermit programs perform CRC error checking.

Default mode

Select **Text** or **Binary** as the default transfer mode (used only for Kermit and Zmodem protocols). When you select Upload or Download from the main menu, the default mode will be selected. Also, select whether to overwrite existing files when downloading. You can change any of these options in the Upload or Download panel before performing the transfer.

Default download directory


Enter the drive and directory where you want received files to be stored by default. You can change this directory in the Download panel before performing the transfer. If you set the “Automatic Zmodem download” option, then this is the directory where the received files are stored.

Automatic Zmodem download

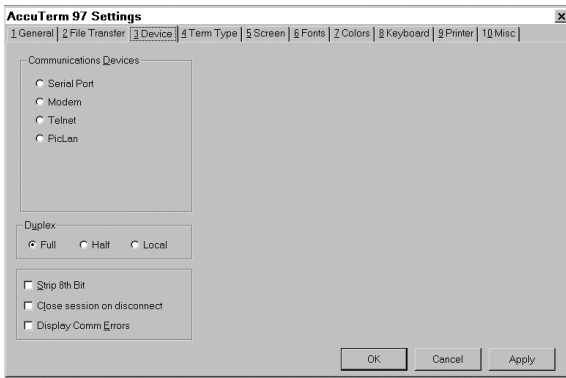
If this option is set, then AccuTerm will automatically begin receiving files using the Zmodem protocol when the other machine begins sending files using the Zmodem protocol. The files are saved in the “Default download directory”.

Session Settings

Each AccuTerm session has settings that define the terminal being emulated, the communications device, port, baud rate, etc. Once a session has been started (by **File New** or **File Open** from the main menu), you're ready to adjust the session settings to be compatible with the host computer with which you desire to communicate.

The *session Settings* are organized into 8 categories: **Device**, **Term Type**, **Screen**, **Fonts**, **Colors**, **Keyboard**, **Printer** and **Misc**. To change settings for a session, activate the session which you want to configure. Then pull down the **Tools** menu and select **Settings** or click the  button on the toolbar.

Device Settings



Communications Device

The **Communications Device** group of option buttons is used to select a communications device type. Once a device type has been selected, other configuration options will become available. For example, if **Serial Port** is chosen, various options for a serial port, such as port number, baud rate, etc. will be displayed. The settings for each device type will be described later in this section.

Duplex

There are three duplex options: **Full**, **Half** and **Local**. In full duplex, all characters typed on the keyboard are sent to the host computer. They are not printed on the screen. The host usually sends (echoes) the same character back; this is what is displayed on the screen. This is by far the most common method. In half duplex, the characters are sent both to the host machine and displayed on the screen. In local mode, characters are displayed directly on the screen and are not sent to the host computer.

Strip Eighth Bit

Most communications devices use eight bits to represent each character. When using eight bits, the computer can send up to 256 different character codes. Some emulations have no way of handling characters above 127 (hex 7F). In this case, you can check this box and the eighth bit of each byte will be automatically stripped from all received characters. If you need to use any international characters or special symbols from the PC character set, you need to clear this box to access them (the Wyse 60 and VT220 emulations have commands to access these characters when using only 7 bits).

Close session on disconnect

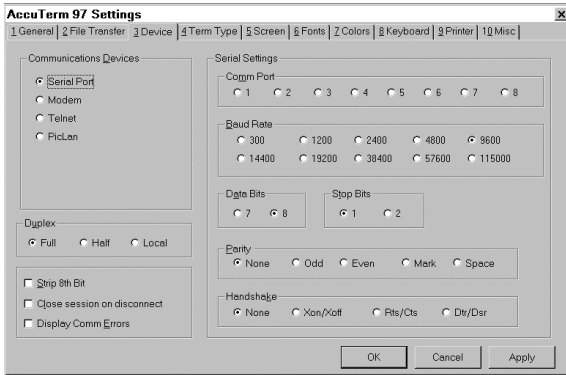
This option is most useful when using Telnet (TCP/IP) connections. If this option is checked, when the session is terminated by the host computer, the session will automatically close. This option may be used in conjunction with the global “Close AccuTerm when all sessions closed” option.

Display Comm Errors

This option determines if communications errors are displayed or ignored. When using network connections, it is better to show communications errors so that they can be fixed; for serial connections, “UART Overrun” errors are common and not serious, so disabling this option is recommended. If you suspect you are having errors and do not have this option set, try setting this option, then resetting the terminal (Select the **Reset Connection** option from the main menu).

Device Settings

Serial Port



Serial Port Device

AccuTerm supports up to 8 serial ports at baud rates up to 115,000 baud. AccuTerm uses the Windows device driver to access the serial ports. Because of this, AccuTerm is limited by the capabilities of the driver. To define the I/O addresses or IRQ numbers for serial ports, you need to use Windows Control Panel, System, Device Manager, Ports, Properties.

Comm Port

This frame is where the desired serial port is selected. AccuTerm shows eight serial ports, however, most systems only have two ports.

If you have more than two ports, there are special problems that you should be aware of. On many systems, serial ports 3 and 4 share interrupts with ports 1 and 2. This can cause problems if you are trying to use both ports 1 and 3 or ports 2 and 4 at the same time. Because of this, we do not recommend that you use them together. This effectively restricts AccuTerm to 2 sessions when using only serial ports (1 session if you have a serial mouse on port 1 or 2).

There are some exceptions: if serial ports 3 and/or 4 are assigned to unique interrupts (not IRQ3 or IRQ4) you should be able to use the additional ports. Also, if you have a multiport serial card with a suitable Windows driver, you can have up to eight serial sessions.

Baud Rate

Baud rate is the speed that the serial port communicates with the host or modem. AccuTerm supports baud rates from 300 to 115,000 baud. If your PC is connected directly to the host machine, you will most likely use 9600 or 19,200 baud. The speed that you use depends on the host computer.

Data Bits

This refers to the number of bits the serial port sends to represent the actual character. The options are 7 or 8 bits (the default is 8 bits). This setting must match your host system.

Stop Bits

This is the number of bits that the serial port uses to separate each character sent. The options are 1 or 2 stop bits (default is 1). This setting must match your host system.

Parity

Parity is an error checking mechanism performed by the serial port. The options are: None, Odd, Even, Mark and Space. The most common is None (this is the default). This setting must match your host system.

Handshake

Computers can send information very quickly. But there are times when the information cannot be processed by the PC as fast as the host can send it. In these cases AccuTerm must have a way to tell the host computer to slow down. This process is called handshaking.

AccuTerm supports three methods of handshaking. The first and most common is. Xon/Xoff. With this method, AccuTerm sends a special Xoff character (**DC3**) to the host system when it wants the host to stop sending characters. When the AccuTerm has caught up, it sends an Xon character (**DC1**). Because this method is interpreted by the software on both machines, it is often referred to as *software handshake* protocol. AccuTerm uses only inbound Xon/Xoff handshake. That is, AccuTerm can control flow of data from the host, but the host cannot control the flow of data from AccuTerm.

The second method is DTR/DSR. This method is referred to as a *hardware handshake* protocol because it physically changes the voltages on the serial

connector. In normal operation, the DTR pin of the serial connector is held high (has a positive voltage applied). If AccuTerm needs to stop the host computer from sending data, it drops the DTR signal (negative voltage applied).

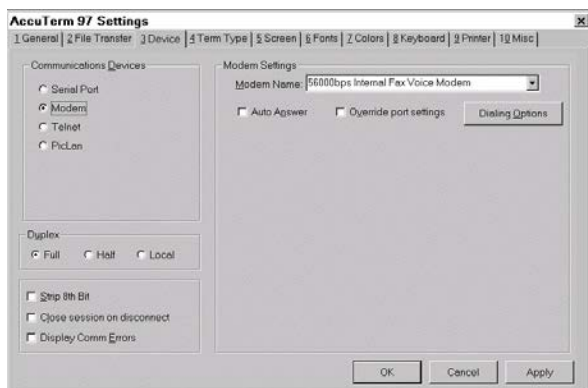
The third method, RTS/CTS, protocol works the same as DTR/DSR except it uses a different pair of signals (pins) on the serial port.

For the DTR/DSR or RTS/CTS protocols to work, the cable which connects the PC and the host computer must connect the appropriate handshake signals at both ends. Physical cable wiring is beyond the scope of this manual.

This setting must match your host system.

Device Settings

Modem



AccuTerm 97 supports dial-up connections using modems which have been installed on your system. AccuTerm uses TAPI (a service built into Windows for supporting telephony) to communicate with your modem. By using TAPI, AccuTerm can share the modem with other TAPI enabled applications, like Windows fax applications.

Modem Name

To use a modem connection, select a modem from the Modem Name list.

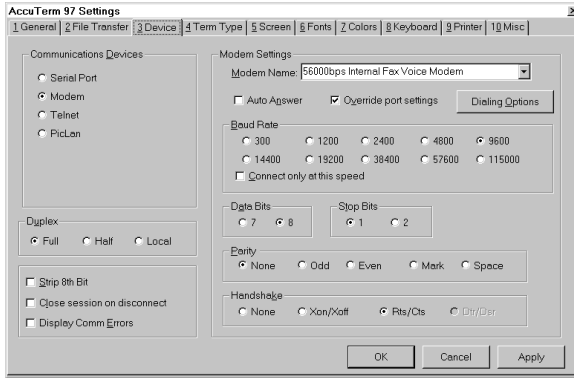
Auto Answer

If you want AccuTerm to answer incoming calls, check the “Auto Answer” setting, otherwise clear this setting.

Override Port Settings

Normally, AccuTerm configures the modem based on settings that you set up in Control Panel. Checking this option allows you to override those settings. When this option is checked, options for selecting the modem baud rate, data bits, stop bits, parity and handshake become available.

See the previous section for details about baud rate, data bits, stop bits, parity and handshake settings. When the “Override port settings” option is selected, the following screen is displayed:



Dialing Options

To modify the dialing properties for the selected modem, such as your current location, dialing prefixes, etc., click the “Dialing Options” button.

Connect Only at This Speed

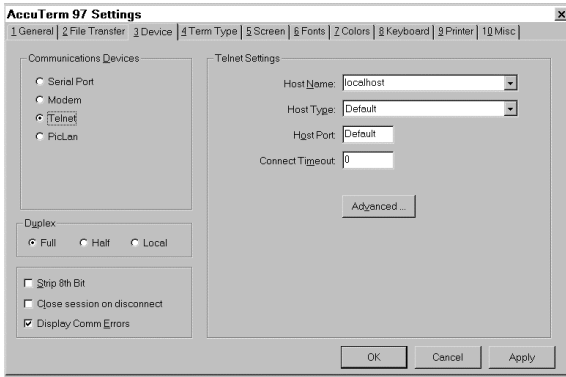
Normally, the baud rate of the modem’s COM port is set to the speed selected in the baud rate setting. However, the baud rate used to communicate between your modem and the computer’s modem is negotiated by the two modems, and may not be the same as the baud rate setting. By checking the “Connect only at this speed” option, AccuTerm will attempt to lock the modem-to-modem baud rate at the same speed as the baud rate setting.

This option is useful if you are able to dial and connect to the host computer modem, but the characters you receive are unintelligible garbage. This usually indicates a mismatch in the baud rate, usually between the host computer and its modem.

Please note that this option does not work with all modems. You can determine if your modem supports this option by opening Control Panel, double-click on Modems, select your modem and click on Properties. If the “Only connect at this speed” option is available (i.e. not gray) then your modem supports this option.

Device Settings

Telnet



AccuTerm supports Telnet connections to host machines on a TCP/IP network. Any Telnet server on the network (including hosts on the Internet) can be accessed.

When using AccuTerm with a network connection such as Telnet, it is recommended that the **Display comm errors** option be set. If this option is not set and a network error occurs, you will not see any indication of the error; the terminal session may appear to hang.

Host Name

A valid Telnet host name or IP address must be entered in this field. If a hosts file is available, you can click on the down-arrow to display a list of available Telnet hosts.

Host Type

If your host is listed in the Host Type list, select it, otherwise just use the Default host type. By selecting a host type from this list, certain advanced options are set up to be compatible with the specified host type. To see the advanced options, click on the Advanced button.

Host Port

If the Host Port field is left blank, or set to “Default” AccuTerm will use TCP/IP port 23 (the default port for Telnet on TCP/IP). If a different port is used for Telnet on your network, enter that port number in this field.

Certain host systems, such as Pick D3, allow a host process number (PIB) to be mapped to a specific TCP/IP port number. You can use the Host Port setting to specify the port number mapped to your process number. Consult with your system administrator for this setting.

Connect Timeout

When establishing a connection to a host computer, AccuTerm will wait up to the specified number of seconds before giving up. If this field is left blank, or zero, the default timeout is 15 seconds.

Advanced Options

By clicking on the Advanced... button, you can modify various Telnet options. These options are set by selecting a Host Type, but may be modified.

Echo (advanced option)

Most systems require the host to echo characters received from the client (terminal). That is, when you press a key, the character is sent to the host, and the host sends the same character back to the terminal, at which time it is displayed. If this behavior is desired, then select the echo option (default). If the echo option is not selected, then the **Duplex** setting should be changed to “half” to enable local echo of typed characters.

Binary (advanced option)

When the binary option is selected (default), data is transferred between the host and terminal without modification. This is useful when transferring binary data such as graphics. When this option is not selected, the **CR** characters not followed by **LF** may be sent as **CR NUL** instead. *Note: when this option is selected, some servers expect lines to be terminated with **LF** instead of **CR**.*

Use alternate break signal (advanced option)

When this option is selected, pressing the **CTRL+BREAK** sends the Telnet Break command. Otherwise the Telnet Interrupt Process command is sent (default).

Bypass option negotiation (advanced option)

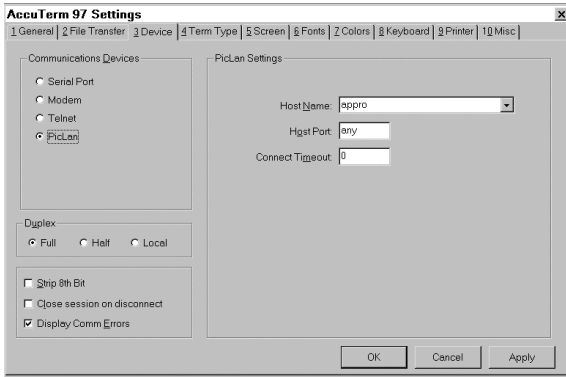
When this option is selected, Telnet option negotiation which occurs when establishing a connection is skipped. Certain host systems, like Pick's D3, which do not support Telnet option negotiation require this option.

Terminal Type (advanced option)

Many Telnet servers allow the client (terminal) to specify a terminal type when a connection is established. AccuTerm uses the default terminal types of *tty*, *viewpoint*, *viewpoint60*, *wyse50*, *wyse60*, *vt220*, *vt52*, *ansi* and *pickmon*, depending upon the selected terminal emulation. If a different terminal type is required, enter that type in this field.

Device Settings

PicLan



AccuTerm supports Modular Software's PicLan network for Pick hosts. Any Pick host (server) on the PicLan network can be accessed.

In order to use a PicLan connection, AccuTerm requires the 32 bit version of PicLan using PLAN32.DLL.

AccuTerm supports up to ten PicLan connections allowing up to ten concurrent PicLan sessions.

When using AccuTerm with a network connection such as PicLan, it is recommended that the **Display comm errors** option be set. If this option is not set and a network error occurs, you will not see any indication of the error; the terminal session may appear to hang.

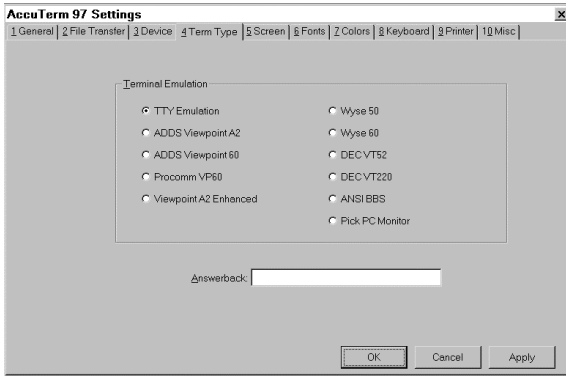
Host Name

A Pick host (server) name must be entered in this field. Click on the down-arrow to display a list of available Pick hosts (AccuTerm queries the network to assemble this list, it may take several seconds for this to occur).

Host port

Enter a specific port number used for connecting to the Pick host. This field may be left blank, signifying "any" available port (default). Because some applications and/or environments require the use of a fixed port number for security or other purposes, a specific port number may be entered here.

Terminal Type Settings



This section allows you to choose the type of terminal to emulate and set the terminal ID (Answerback). The emulation you select depends on your host computer software. If you are not sure which terminal type to use, try different ones until the screen looks right.

Terminal Emulation

Most of the emulations are self explanatory. **Procomm VP60** emulates the ADD5 Viewpoint 60 terminal the same way Procomm does (which is different from a true Viewpoint 60 terminal). The **Viewpoint A2 Enhanced** setting emulates the Wyse 50 ADD5 A2 personality with *enhance* turned on. **Pick PC Monitor** emulates the console port of the Pick system (R83 and Advanced Pick).

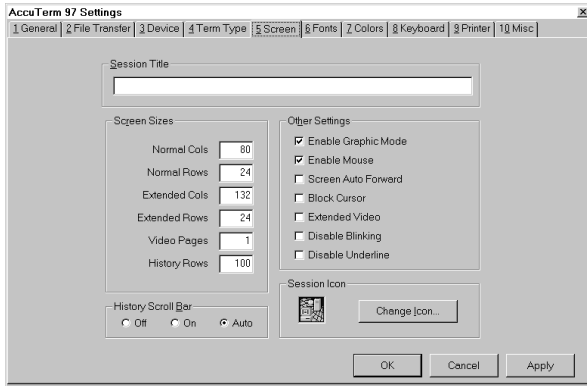
Besides the terminal types listed on this screen, AccuTerm also emulates the Tektronix 4010/4014/4100 graphics terminal. The Tektronix emulation is selected with a software command while running any of the other emulations.

Answerback

The contents of this field are sent to the host when AccuTerm receives a request to transmit the answerback string under Wyse and VT220 emulations. When the answerback message is sent to the host under Wyse emulation, an **ACK** control character follows the message. Under VT220 emulation, no terminator is sent.

Control characters may be entered into the answerback field by prefixing the character with a caret (^).

Screen Settings



This category lets you set the terminal screen options like the size, cursor type, etc. You can also change the session title or the Icon used when the session is minimized.

Session Title

This is the text that will be displayed on the session window title bar. If the title is blank, the name of the current configuration file is displayed. When the session is maximized, the session title becomes part of the AccuTerm main window title.

Screen Sizes

There are two separate screen sizes that AccuTerm supports; normal and extended. These sizes are used to emulate the 80 and 132 column modes of the Wyse and VT220 terminals. AccuTerm has taken the extended and normal modes one step further. With AccuTerm you can define both the normal and extended modes to be any number of columns or rows up to a maximum of 160 columns by 60 rows.

You will probably leave these settings at their default settings of 80 characters by 24 rows for normal mode and 132 characters by 24 rows in extended mode. If you have software that supports other screen sizes, you can experiment with these settings until you find a size that you like.

AccuTerm always allocates one extra row for function key labels, host messages, etc. If you specify 24 rows here, AccuTerm will display 24 data lines and one message line (the Pick PC Monitor and ANSI BBS emulations display 25 data lines).

When AccuTerm switches between the normal and extended modes, one of two things will happen: either the characters will change size to fit the visible screen, or scroll bars will be displayed if the full screen will not fit within the session window at the selected character size. For more discussion on scaleable fonts, see the section on Fonts.

Video Pages

Some of the terminals AccuTerm emulates support multiple addressable pages. With AccuTerm, you can have up to 25 pages. Although most software doesn't make use of this number of pages, you can use them with the Screen Auto Forward feature discussed below. Each page requires memory, so if you don't use them, it is better to leave this number at the default of 1.

History Rows

AccuTerm supports a "history buffer", which saves lines which are scrolled off of the terminal screen. To use this feature, specify how many rows (up to 1200) are to be saved. When the history buffer has been enabled (by setting the number of rows), use the vertical scroll bar, mouse wheel, or keyboard commands to scan history.

You can use **CTRL + SHIFT + PGUP**, **CTRL + SHIFT + PGDN**, **CTRL + SHIFT + UP**, **CTRL + SHIFT + DOWN**, **CTRL + SHIFT + HOME**, and **CTRL + SHIFT + END** to scan the history buffer.

The history buffer can be “locked”. Once locked, lines which are scrolled off of the terminal screen are not saved in the history buffer. This enables you to capture important information in the history buffer, lock the contents, and proceed with other work with the ability to refer to the history at any time.

The history buffer can be copied to the clipboard; either the entire history buffer may be copied (**E**dit **C**opy **H**istory menu command), or you can select a block to copy using the mouse, then use the **E**dit **C**opy menu command.

History Scroll Bar

When a history buffer has been enabled, the vertical scroll bar may be used to scan through the history. AccuTerm supports three scroll-bar settings.

Auto: when “Auto” (the default mode) is selected, placing the mouse near the right edge of the screen will cause the scroll bar to “slide” into place. Moving the mouse away from this area causes the scroll bar to disappear.

On: when “On” is selected, the scroll bar will always be present.

Off: when “Off” is selected the scroll bar will only be present if the window height is not large enough to display normal number of rows.

Note: when Auto or Off are selected, using the keyboard commands or mouse wheel to scan history causes the scroll bar to be displayed briefly. This is so that you can see the position of the scroll button after scrolling the window.

Other Settings

Enable Graphic Mode

Checking this box enables the Tektronix 4010/4014/4100 graphic terminal emulation. If the graphics mode is enabled, AccuTerm will automatically switch to the Tektronix graphic emulation when it receives the appropriate command from the host computer.

There may be times when you want to disable the graphic emulation. If you have a bad modem connection and are getting random garbage characters on the screen, AccuTerm may accidentally switch into graphics mode. Sometimes a software program may send the sequence at the wrong time. If this happens, disable the graphics mode until you need it.

Enable Mouse

This option determines whether host mouse support is enabled. Under normal operation, if you click on the terminal screen with the mouse, nothing happens. If a "mouse on" command is received from the host while this option is checked, AccuTerm sends a special code informing the host of the mouse location when the button was pressed, and which mouse button was pressed. The mouse cursor is also changed to a "pointing hand" to indicate that the mouse is active.

Screen Auto Forward

When this field is checked, it turns on the Screen Auto Forward feature. This feature works in conjunction with the video pages to provide a "screen back" buffer. Whenever the host computer sends a clear screen command, AccuTerm switches to the next page before clearing the screen. This saves the prior page.

To look at the previous page, press **CTRL + PGUP**; to look at the next page, press **CTRL + PGDN**.

Block Cursor

Checking this option turns the cursor into a blinking block instead of an underline.

Extended Video

Checking this option sets the screen to extended (132 column) mode. This can be used to manually switch between extended and normal (80 column) modes.

Disable Blinking

Checking this option disables the "blinking" visual attribute.

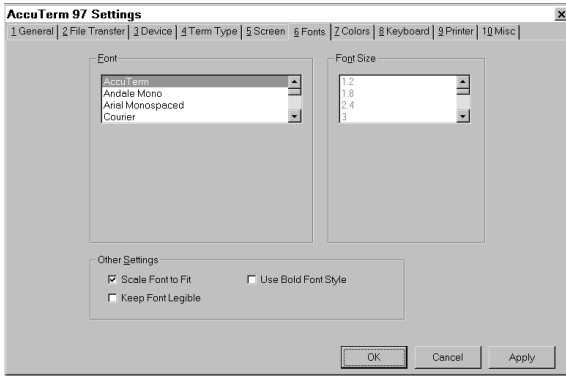
Disable Underling

Checking this option disables the "underline" visual attribute.

Session Icon

Any terminal session can be minimized (click the minimize button in the session title bar or select **Minimize** from the system menu (**ALT + -**)). All minimized sessions are shown as icons at the bottom of the AccuTerm main window. To change the icon used to represent the current session, click this button or press **ALT + I**, then select an icon file (files ending in *.ico*) to represent this session when it is minimized.

Font Settings



This category is used to select the type and behavior of the screen fonts.

Font

This list shows all of the available fonts installed on your computer. There are two types of fonts under Windows: fixed pitch and proportional. Proportional fonts take different amounts of space for each character depending on the width of the character. For example, the letter "I" takes up less space than the letter "W". Fixed pitch fonts take up the same amount of space for each character. AccuTerm only lists fixed pitch fonts.

The first font listed is the **AccuTerm** font. This font has been optimized to look good over a wide range of character sizes and screen resolutions. We recommend that you use this font, especially if you have a low resolution screen (640 x 480).

Font Size

This box lists all font sizes available for the font selected in the **Font** list. The **Font Size** list is disabled if the **Scale Font to Fit** option is checked.

Scale Font to Fit

When this option is checked, AccuTerm automatically re-sizes the selected font so that the entire terminal screen is visible. This allows you to re-size the screen or change between normal and extended video mode without having to deal with scroll bars to see everything. However, if for some reason, you need lock the characters at a specific size, you can un-check this setting and then pick the size you want from the **Font Size** list.

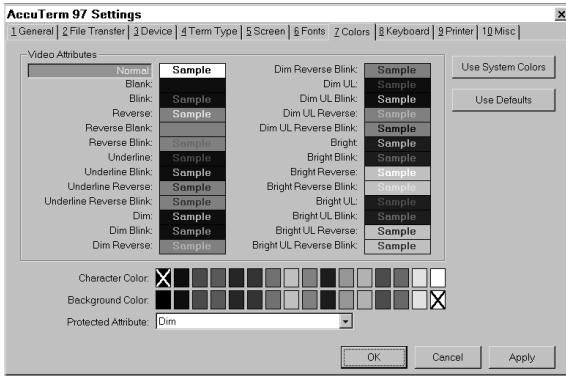
Use Bold Font Style

If this option is checked, AccuTerm uses the **Bold** font style if it's available in the selected font.

Keep Font Legible

If this option is checked, and the **Scale Font to Fit** option is checked, AccuTerm will never scale a font smaller than 6 pixels wide by 9 pixels high.

Color Settings



Windows is capable of displaying text in a wide variety of colors. The colors are separated into character color and the background color. Most of the terminals being emulated do not have color commands. Instead, they have video attributes like normal, reverse, dim, blink, underline, etc. This category is used to map the various terminal video attributes into a combination of character and background colors.

This panel is also used to modify the palette of colors used by AccuTerm. AccuTerm uses a palette of 16 colors. Since most systems are capable of displaying more than 16 colors, AccuTerm allows you to customize its palette by using the “System” color settings (Control Panel, Display, Appearance) or by selecting a color using the color selection dialog box. To use System colors, click the **Use System Colors** button. To open the color selection dialog box, double-click on one of the palette colors shown at the bottom of the panel. To restore the palette to AccuTerm’s default colors, click on the **Use Defaults** button.

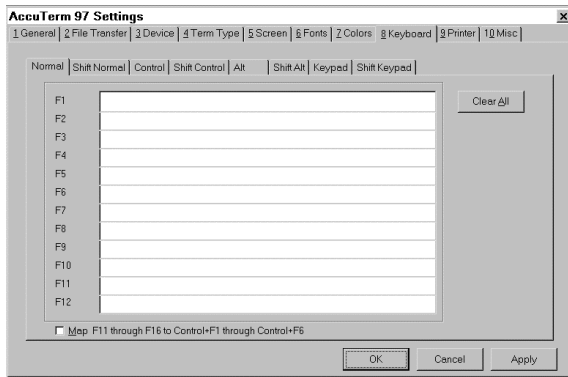
To change the color mapping, first click on the **Video Attribute** whose colors you want to modify. The sample box to the right of the attribute name shows the current color combination selected for that attribute. Next, click on the desired character and background colors in the palette shown at the

bottom of the dialog box. The selected character and background colors are indicated by a large X in the palette.

The Wyse 50 and Wyse 60 emulations require a video attribute to be assigned as the “Protected” attribute. To select the **Protected Attribute**, click on the drop-down arrow, then select which video attribute should be used as the default protected attribute.

Once you have the colors set the way you want them, press **ENTER** or click **OK** to return to the session and see how they look.

Keyboard Settings



This category is used to manually reprogram the function (**F1** to **F12**) and editing keys. The keys can be programmed with a character sequence which is transmitted to the host when the key is pressed, or a macro command, which is executed when the key is pressed.

To program a key with a sequence of characters, enter the sequence into the field next to the key name. To enter control characters in a key sequence, prefix the control code with a caret (**^**). For example, to enter a **CTRL+J** (the line feed character) you would enter **^J**.

Certain control characters do not have a corresponding letter key; to program the **ESC**, **FS**, **GS**, **RS**, **US** or **DEL**, control codes, enter **^[**, **^/**, **^]**, **^~**, **^_** and **^?** respectively. To program a caret (**^**), enter two carets (**^^**). To program a trailing blank character, enter a single caret after the trailing blank. Characters with the eighth bit set are programmed by preceding the 7 bit character with a backslash (****). For example, to program a **CHAR(205)**, enter **\M** (**M** is **CHAR(77)**; **** indicates the eighth bit is set (**128**; **77+128=205**). To program a backslash (****), enter two backslashes (****).

For example, to program the **END** key to send the word "END", followed by a carriage return, first select the **Keypad** tab, then in the field next to the **End** label, enter the sequence:

E N D ^ M

To program a key with a macro, enclose the macro command in square brackets ([]). See the **Scripting** section for more information on macros.

Most of the emulations have a default sequence which is transmitted when the function or editing keys are pressed. When a key is not programmed, it will send the default sequence (see the **Appendix** for default sequences). Keys which are programmed using this panel override the default sequence, unless they are re-programmed by the host system to their default settings.

The programmable keys are divided into eight categories: **Normal**, **Shift Normal**, **Control**, **Shift Control**, **Alt**, **Shift Alt**, **Keypad** and **Shift Keypad**. The **Normal** key sequence is sent when one of the function keys is pressed by itself. The **Control** sequence is sent when a function key is pressed while holding down the **CTRL** key. The **Alt** sequence is sent if the **ALT** key is held down while pressing a function key. The **Keypad** sequence is sent when one of the editing keys is pressed. The **Shift Normal**, **Shift Control**, **Shift Alt**, and **Shift Keypad** sequences are sent when the **SHIFT** key is held down in combination with any of the previous keys.

The Keypad category is used for the special editing keys **BKSP**, **TAB**, **INS**, **DEL**, **HOME**, **END**, **PGUP**, **PGDN**, **LEFT**, **RIGHT**, **UP**, and **DOWN**.

To manually program function and keypad keys, you must first select the **Key Type** from the tabs at the top of the panel. Next, enter the characters you want the programmed key to send in the appropriate field below.

Printer Settings



AccuTerm 97 uses the printer in two ways: First, AccuTerm can print the contents of the terminal screen by using **File Print Screen** menu command (or toolbar button), or by receipt of the appropriate command from the host computer.

The other way AccuTerm uses the printer is to emulate a “slave” printer attached to a terminal’s Aux, or printer port. This mode is enabled by using the **File Printer On** menu command, the **ALT + P** keyboard command, or by receipt of the Printer On command from the host. The state of the “Aux printer” is indicated on the status bar (Idle, Off or On).

When AccuTerm receives a Printer On command from the host (or by using the **File Printer On** menu command or **ALT + P** keyboard command), the printer status shown in the status bar is changed to “On”. When the Printer Off command is received, the status changes to “Off”. If another Printer On command is not received before the number of seconds specified in **Printer Close Timeout**, the print job is closed, and the status changes to “Idle”.

AccuTerm can print in two modes: text and graphics. Text is the mode used by “real” terminals. This mode is different from normal Windows printing in that the Windows printer driver is not used. For screen printing, text mode

sends the screen text to the printer. For slave printing, AccuTerm passes raw data, including control characters and escape sequences from the host directly to the printer (or spooler). In text mode, the host computer is responsible for setting printer font sizes and page orientation by sending the appropriate printer commands.

In graphics mode, AccuTerm prints an image of the screen on the printer, using the Windows printer driver. When the screen is printed in graphics mode, the printed image includes text, line drawing characters, and may optionally include the screen background and colors. For slave printing, AccuTerm understands **CR** (carriage return), **LF** (line feed) and **FF** (form feed) control characters, but other control characters and escape sequences are ignored. You can select the font, paper size and page orientation used for graphics mode slave printing by clicking the Page Setup button in the printer settings panel.

Printer Selection

By default, AccuTerm uses the system default printer (the **Use Windows Default Printer** option selected). If you want to select a different printer, select the **Use Specific Printer** option, then select a printer from the drop-down list. If you want to disable printing from AccuTerm, select the **Disable printing** option.

Screen Print Settings

AccuTerm can print the screen in **Text** or **Graphics** modes. In **Text** mode, only the text characters currently displayed on the screen are printed. In **Text** mode, you can elect to **eject** the page from the printer before the printout, after the printout or both.

In **Graphics** mode, an “image” of the current screen is printed, including special symbols like line drawing characters. You can elect to print the **background** (shading and highlighting) or print the screen in **color** if you have a color printer.

Slave Printer Settings

AccuTerm can print to the “slave” or “aux” printer in either **Text** or **Graphics** mode. In **Text** mode, you can **elect** to eject the page from the printer before the printout, after the printout or both. In Graphics mode, you can select the font, page orientation, paper size and paper source by clicking the **Page Setup** button.

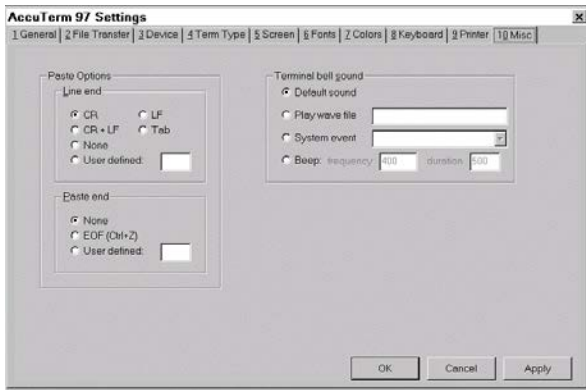
Printer Close Timeout

Enter the number of seconds to wait after receiving a Printer Off command before closing the print job. Usually, when the host computer needs to print a report, it usually sends a Printer On command, sends the printer data and then sends a Printer Off command. But some host programs send a Printer On command, send some data to the printer, send a Printer Off command, and send some other data to the screen, and then send another Printer On command, and send more data to the printer, etc. Because of this, AccuTerm has no way of knowing when the host is finished sending the print job. The only way it can determine this is to wait for a set amount of time after the Printer Off command before closing the job. If this field is set to the default of 5 seconds, AccuTerm waits 5 seconds after the Printer Off command has been received before closing the print job.

This causes a delay when printing under AccuTerm. If all your host programs always turn the aux port on, send all the data and then turn it off, you can reduce this number to 1 or 0. If you have programs that are constantly turning the aux port on and off, you may need to adjust this number until things work correctly.

Miscellaneous Settings

The Miscellaneous Settings category is used for setting other Session options, such as how to handle paste operations and to customize the terminal bell sound.



Paste Options: line end

When the clipboard is “pasted” to an AccuTerm session, AccuTerm transmits the clipboard text to the host computer. The **line end** options specify what is transmitted at the end of a line of text. Options include **CR** (default), **LF**, **CF+LF**, **TAB**, none or a user-defined character. To specify a user-defined character, select the **User defined** option and enter the ASCII code of the character to be transmitted at the end of each line.

Paste Options: paste end

After pasting the clipboard to an AccuTerm session, AccuTerm can transmit an optional **paste end** character. Normally, AccuTerm does not transmit any character at this time, but you can select **EOF** (which is the **SUB** control code), or user-defined. To specify a user-defined character, select the **User defined** option and enter the ASCII code of the character to be transmitted at the end of a paste operation.


Terminal Bell Sound

The sound generated by AccuTerm upon receipt of the ASCII **BEL** control code can be customized. The “default sound” depends upon whether a sound card is installed or not. If a sound card is not installed, a “beep” is produced; otherwise the sound defined by the Control Panel Sounds applet as the “Default Beep” is played.

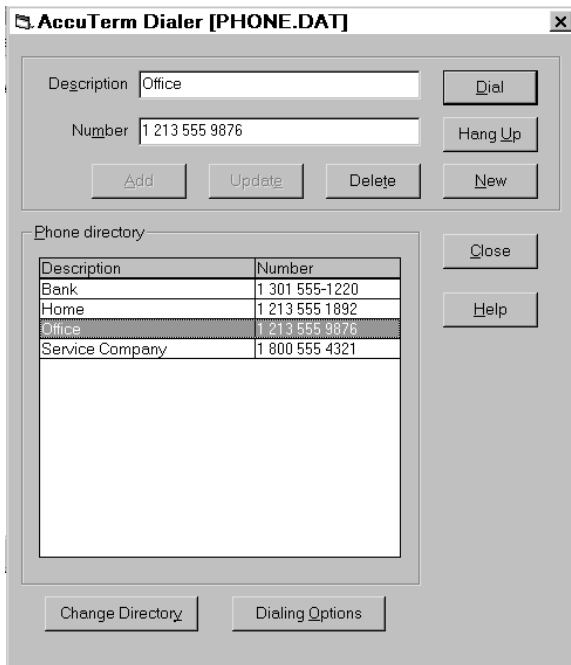
Other options which may be selected are: play wave file, system event or beep. To play a wave file, select this option and enter the full path for the wave file to be played. To use a sound associated with a system event, select the event from the dropdown list. System events are associated with wave files in the Control Panel Sounds applet. To use a true beep, enter the frequency of the beep in Hz, and the duration in milliseconds. The default is 600Hz for 200ms. The frequency must be between 200Hz and 4000Hz. The duration must be between 100ms and 5000ms.

AUTO DIALER

The AccuTerm Auto Dialer gives you an easy way to store and dial phone numbers. The **Phone directory** list contains all the phone numbers in the current directory file. You can have as many phone numbers in a directory file as you want. You can also have any number of directory files. The default directory file name is set in the **General** category of the **Settings** dialog box (**T**ools **S**ettings menu).

To access the Auto Dialer, select the **T**ools **A**utodial menu item click the  button on the toolbar.

Note: the Auto Dialer is only available if the current session is configured to use a Modem as its communications device.



The dialog box titled "AccuTerm Dialer [PHONE.DAT]" contains the following elements:

- Fields for "Description" (containing "Office") and "Number" (containing "1 213 555 9876").
- Buttons: "Dial", "Hang Up", "Add", "Update", "Delete", "New", "Close", and "Help".
- A "Phone directory" section with a table:

Description	Number
Bank	1 301 555-1220
Home	1 213 555 1892
Office	1 213 555 9876
Service Company	1 800 555 4321

Below the table is a large empty rectangular box. At the bottom of the dialog are buttons for "Change Directory" and "Dialing Options".

Dialing

Using the Auto Dialer is very simple. To dial a number, double click an entry in the phone directory, or highlight the desired number in the phone directory list then press **ALT + D** or click the **Dial** button.

If you want to dial a number which is not in the directory, simply enter the desired phone number in the **Number** field, then click **Dial**.

Hanging Up

To terminate a connection, press **ALT + U** or click the **Hang Up** button.

Adding a new entry

To add a new entry to the directory, press **ALT + N** or click on the **New** button. Enter a phone number and optional description, then press **ALT + A** or click the **Add** button.

Deleting an entry

To delete the selected entry, press **ALT + T** or click the **Delete** button.

Modifying an entry

To change an existing description or number, select the entry you want to change, modify the description or number and press **ALT + E** or click the **Edit** button.

Creating a new directory file

To create a new phone directory file, click the **Change Directory** button. From the **Open Phone Directory File** dialog, select the drive and directory for the new phone directory file, and enter the new phone directory file name. Click **OK** to create the new phone directory file.

Changing directory files

To select a different phone directory file, click the **Change Directory** button, then select the new phone directory file. Click **OK**.



Changing dialing options

To change the dialing options such as your current location, dialing prefix, credit card information, etc., click the **Dialing Options** button.

Directory file format

Phone directory entries are stored in a standard tab delimited ASCII file with a file extension of *.dat*. The name is the first field and the phone number is the second field. Each entry has it's own line. Many applications have the ability to export data to ASCII files so it may be possible to use your existing phone directories from other applicatoins with the Auto Dialer.

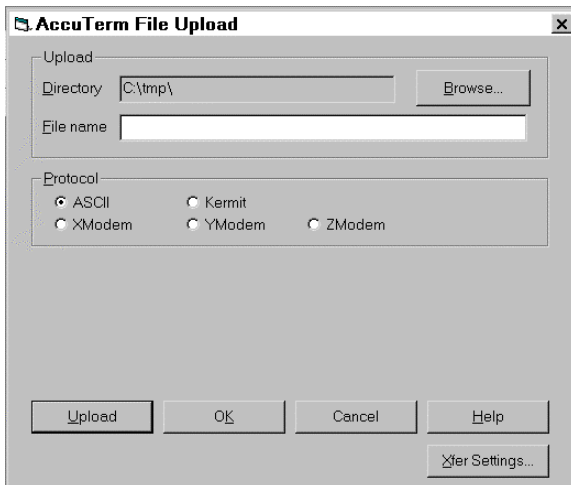
FILE TRANSFER

AccuTerm has the ability to transfer files between your workstation and a remote machine using one of the five built in file transfer protocols: ASCII, Kermit, Xmodem, Ymodem and Zmodem. The file transfer features are accessed from the **T**ools menu by selecting the **U**pload or **D**ownload items or by clicking on the  or  buttons on the toolbar.

In order for the file transfer to work correctly, you may need to adjust certain protocol parameters. These are found in the **F**ile **T**ransfer category of the **S**ettings dialog box (**T**ools **S**ettings menu).

*Note: file transfers between AccuTerm and Pick (or Pick-like) host systems are more easily accomplished using the Pick file transfer programs included with AccuTerm - see **Appendix E** for more information on the Pick file transfer programs.*

File Upload



Upload transfers a file (or files) from the workstation to a remote machine. Uploads may use any of the supported protocols. Usually, to perform a file upload, you would first start the receiving (download) program on the remote machine while in normal terminal emulation mode. Then after the remote receiving program has been started, select **T**ools **U**pload, select the appropriate protocol, select the file(s) to upload, then press **ALT** + **L** or click the **U**pload button.

If you want to select items to upload, but not initiate the transfer immediately, click the **OK** button instead of the **U**pload button. In this case, all of your selections remain until the next time you open the **U**pload panel.

File Selection

Click the **B**rowse button to select the file(s) to upload. If you have selected a protocol which can upload more than one file (**K**ermit, **Y**modem, or **Z**modem), you can select multiple files, otherwise select only a single file.

Protocol

Select the protocol to used for uploading the selected files by clicking the appropriate option button: **A**SCII, **K**ermit, **X**modem, **Y**modem or **Z**modem.

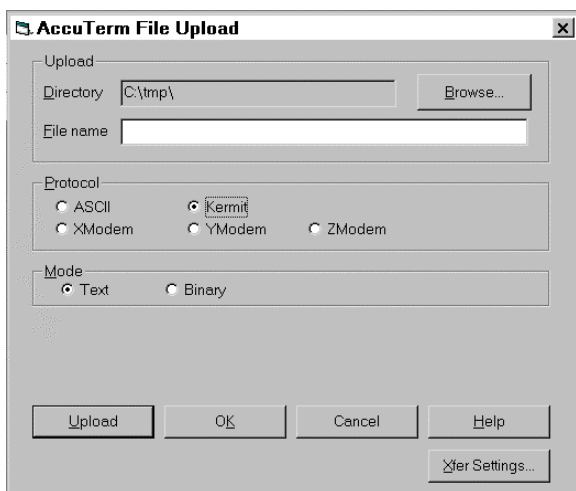
The **A**SCII protocol by its nature has no error checking. This protocol should only be used when there is a local, error free connection, or when the host system does not support any of the other protocols. The **A**SCII protocol only sends text files, and it is not a “batch” protocol - you can only send one file. You can adjust the line termination to **C**R, **L**F or **C**R **L**F by selecting the desired option in the **F**ile Transfer category of the **S**ettings dialog box (**T**ools **S**ettings menu). You can also set the inter-line delay time in the same section (this is the amount of time AccuTerm waits between lines when sending a file).

The **K**ermit protocol provides error detection and correction. **K**ermit is a “batch” protocol, so you can send multiple files. **K**ermit can send either text or binary files; text files are converted to the format required on the receiving machine. **K**ermit is slower than X/Y/Zmodem, but is the most portable protocol. It often succeeds when the other protocols fail. Several options may be set in the **F**ile Transfer category of the **S**ettings dialog box. These include the retry limit, timeout period, end of line character, eighth bit quoting, data compression and block check type (checksum or CRC).

Xmodem is a binary protocol which includes error checking. **Xmodem** is not a “batch” protocol; you can only send one file using **Xmodem**, and you need to specify the file name to the receiving program. Files transferred using **Xmodem** “grow” to the next multiple of 128 bytes. The **Xmodem** protocol requires an 8 bit data path, and full 8 bit transparency (you cannot use Xon/Xoff handshaking when using **Xmodem**). You can set the **Xmodem** timeout period in the **File Transfer** category of the **Settings** dialog box.

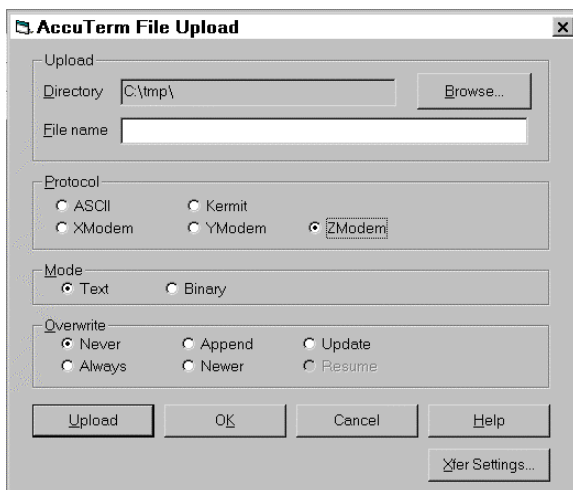
Ymodem is a binary protocol which includes checksum or CRC error checking. **Ymodem** is available in several “flavors”; the “flavor” used by AccuTerm is Ymodem-G (1K blocks, batch oriented). The **Ymodem** protocol requires an 8 bit data path, and full 8 bit transparency (you cannot use Xon/Xoff handshaking when using **Ymodem**). You can set the **Ymodem** timeout period in the **File Transfer** category of the **Settings** dialog box.

Zmodem is a sophisticated file transfer protocol which is popular on PCs and Unix based minicomputers. **Zmodem** provides robust CRC error checking, several options for existing files and text file conversion. If the receiving machine uses the “rz” program, AccuTerm will send the “rz” command before sending the files, to automatically invoke the receiver’s **Zmodem** program. The **Zmodem** protocol requires an 8 bit data path, but full 8 bit transparency is not normally required (you can use Xon/Xoff handshaking when using **Zmodem**). You can set the **Zmodem** timeout period in the **File Transfer** category of the **Settings** dialog box.



Mode

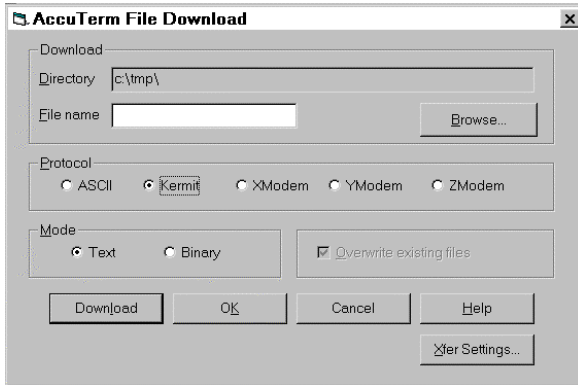
When the protocol is set to **Kermit** or **Zmodem**, a **Mode** frame is displayed. You can select **text** or **binary** mode. When **text** mode is selected the line termination is adjusted to match that of the receiving machine. When **binary** mode is selected, the files are transferred without alteration. **ASCII** always transfers text files, and **Xmodem** and **Ymodem** always transfer in binary mode.



Overwrite

When the Zmodem protocol is selected, another frame, **Overwrite**, is displayed. When using Zmodem, the sender controls the action to take if a file exists on the receiving machine. Select **never** if you want to preserve all existing files. Select **always** to overwrite existing files. **Append** will append onto existing files. The **newer** option replaces existing files only if the file being sent is newer than the existing file. The **update** option overwrites existing files; non-existing files are not transferred. Use the **resume** option for crash recovery after aborted transfers - if a file does not exist on the receiving machine, it is sent; if a file exists, but it is shorter than the file on the sending machine, the remainder of the file is sent; if a file exists, but it is the same length or longer, it is skipped. *Note: the **resume** option may only be used if the transfer mode is binary.*

File Download



Download transfers a file (or files) from a remote machine to your workstation. Downloads may use any of the supported protocols. Usually, to perform a file download, you would start the sending program on the remote machine while in normal terminal emulation mode. Then after the remote sending program has been started, select **T**ools **D**ownload, select the appropriate protocol, select the destination directory (and file name if using **Kermit** or **Xmodem**), then press **ALT + L** or click the **Download** button.

If you want to select the destination directory, but not initiate the transfer immediately, you can click the **OK** button instead. In this case, all of your selections remain until the next time you open the **Download** panel.

Directory and File Selection

Click the **Browse** button to select the destination directory. If you have selected a protocol which requires a file name (**ASCII** or **Xmodem**), then you also need to specify a file name for the downloaded file.

Protocol

Select the protocol to be used for downloading the selected files by clicking the appropriate option button: **ASCII**, **Kermit**, **Xmodem**, **Ymodem** or **Zmodem**. For details about each of the protocols, see the discussion in the previous section.

Mode

When the protocol is set to **ASCII**, the mode is **text**; when it is set to **Xmodem** or **Ymodem**, the mode is **binary**. When using **Kermit**, you can select either **text** or **binary** mode. The mode for Zmodem transfers is determined by the sender.

When **text** mode is selected lines are terminated with a **CR/LF**. When **binary** mode is selected, the files are transferred without alteration.

Overwrite


Check this box to allow existing files to be overwritten. *Note: Kermit and Ymodem always overwrite. When Zmodem is selected, the sender controls the action to take if a file exists.*

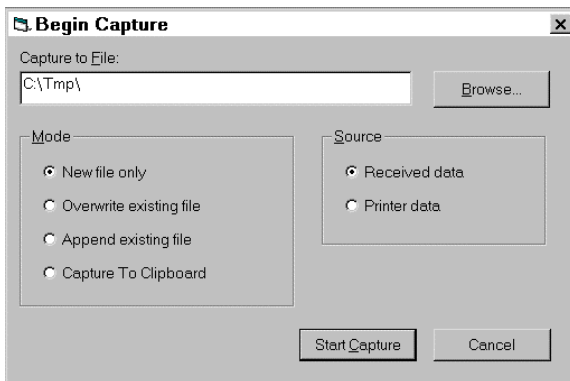
Host Directed File Transfer

AccuTerm has several private escape sequences which may be used to control the file transfer facility from the host computer. See the **AccuTerm Programming** section of this manual for details.

If you are using a Pick-based host computer (Pick, UniVerse, Unidata, etc.), AccuTerm includes a library of Pick/BASIC programs which can be installed on the host. When these host-based programs are used, they greatly simplify the exchange of data between the host computer and the workstation. See **Appendix E** for details on installing and using these programs.

DATA CAPTURE

AccuTerm is capable of capturing to a file any data received by the workstation, or any data which would normally be printed to the slave printer port. To start data capture, select **T**ools from the main menu, then select the **C**apture option or click the  button on the toolbar. Select the capture file and options in the **Begin Capture** panel.



Capture to File

Enter the capture file name in this field, or click the Browse button to select the directory and file.


Mode

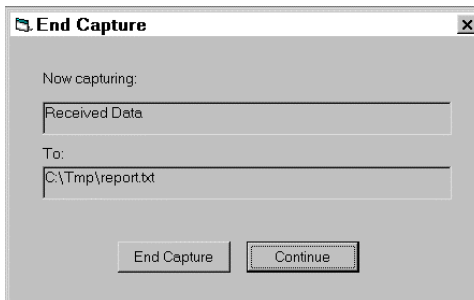
Select **New file only**, **Overwrite**, **Append**, or **Clipboard** by clicking the appropriate option button. If you select **Overwrite** or **Append** and the file does not exist, AccuTerm will create the file. If you select **New file only** and the file does exist, a warning will be displayed.

Source

Select the data source to be captured as **Received data** or **Printed data**. If **Received data** is selected, all data received by AccuTerm (except when a file transfer is occurring) will be captured to the specified file. This includes all control characters. If **Printed data** is selected, all data which would normally be printed to the slave printer port is captured to the selected file, and not printed.

Terminating Data Capture

If data capture is already in effect (as indicated in the status bar), selecting **Tools Capture** or clicking the  button on the toolbar will display the **End Capture** panel. Click the **End Capture** button to terminate the capture operation, or click the **Continue** button to resume capturing.

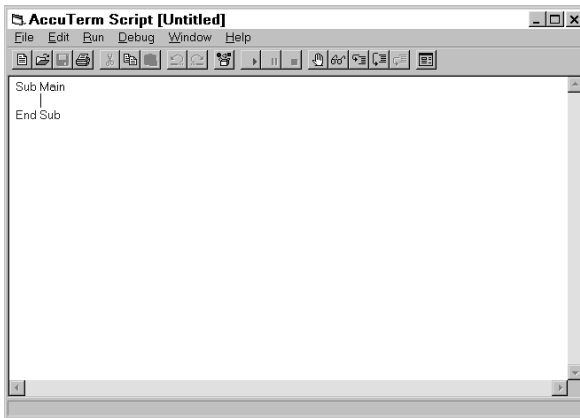


Data capture can also be controlled by the host using AccuTerm private escape sequences. See the chapter on **AccuTerm Programming** for details.

SCRIPTING

AccuTerm includes a powerful scripting language similar to the popular Microsoft Visual Basic Programming System, Applications Edition (VBA). The language has been enhanced to allow your script to control almost every aspect of AccuTerm's operation.

This section describes how to create and use scripts, as well as language extensions. The next section describes AccuTerm's automation interface, which is available to scripts and offers complete control over AccuTerm's operation. For general language details, select the Help menu item from the Script panel; printed documentation is also available from AccuSoft. *This manual does not contain the VBA language reference.*



Creating a Script

Scripts are created by entering VBA statements into the script code window. To activate the script window, pull down the **T**ools menu from the main menu, then select the **S**cript option.

Scripts usually begin with `Sub Main ()` and end with `End Sub`. Enter any legal VBA statements between the `Sub Main ()` and `End Sub` statements.


You can create other subroutines and functions which are called by your `Main()` subroutine (or from function keys, popup menus, host commands, etc). Subroutines begin with the `Sub` keyword followed by the name of the subroutine. If the subroutine requires arguments, enclose them in parentheses. The last statement in a subroutine is `End Sub`.

Functions are similar to subroutines, and begin with the `Function` keyword, and end with the `End Function` statement. Functions return a value; simply assign the return value to the function name within your function code.


Certain declarations must be placed before the `Main()` subroutine. Declare global variables (using the `Global` keyword), DLL functions (using `Declare Function` or `Declare Sub`), and user-defined data types (using `Begin Type` and `End Type`) before `Sub Main()`. Within a subroutine or function, declare any local variables using the `Private`, `Dim` or `Static` statement.

In addition to local references to subroutines and functions defined in your script, those subroutines and functions may also be called from outside of your script in four ways: First, they may be called by script commands sent by the host (see **AccuTerm Programming**). Second they may be called by script commands programmed into function keys (see **Keyboard Settings**). Third, they may be called by script commands executed in response to custom menu and toolbar actions.

Saving a Script File

After the script has been created (or modified), save the script by clicking the  button or pull down the **File** menu (from the script window), and select the **Save** or **Save As** option.

Loading a Script File

To load a script file, click the  button or select the **Open** option from the **File** menu.

Loading and Running a Script File

To load and run a script file, pull down the **File** menu and select **Run**. Enter or select the name of the script file to run, and click the **OK** button. *Note: execution always begins with `Sub Main()`.*

Closing the Script Window

Select the **C**lose item from the **F**ile menu to close the script window. If there are any unsaved changes to the current script, you will be prompted to save changes.





Printing a Script

Select the **P**rint item from the **F**ile menu to print the current script. Select **P**rint **S**etup to select the printer used to print the script.

Editing Scripts

The **E**dit menu provides many editing functions such as **U**ndo, **R**edo, **C**ut, **C**opy, **P**aste, **D**ele~~t~~e, **S**elect all, **I**ndent, **O**utdent, **F**ind and **R**eplace. The font used in the script window may be selected by choosing the **F**ont item, and the dialog editor may be invoked by choosing the **U**ser **D**ialog item.

Running a Script

To run a script from the script window, create a new script or load an existing script into the script window. Pull down the **R**un menu and select **S**tart (Resume) or click the  button. To single step the script, see the topic on **D**ebugging a Script. To terminate the script, select **E**nd from the **R**un menu or click on the  button. To suspend execution, select **P**ause from the **R**un menu or click the  button. To resume execution, select **S**tart (Resume) or click the  button.

Running a Script from the Command Line

To run a script from the command line, append the name of the script file to the target or command line (click the AccuTerm icon with the *right* mouse button, then select **P**roperties). When AccuTerm starts, it loads the script into a hidden script window, and executes its **M**ain subroutine. You can use a command line script to open up a number of sessions, the log a user on, etc.

Running a Script from a Layout File

You can run a script automatically when opening a *layout* file. To save a script with a layout file, open all sessions to be included in the layout file. Open the script window, then load the script you want to automatically run. Return to the main AccuTerm window, and select **File Save Layout**. When the layout file is opened next, the script file will be loaded and its Sub Main() will be executed. *Note: when opening a layout file, all sessions are opened before executing the script.*

Running a Script from a Function Key

To run a script from a function key, simply program the function key (see **Keyboard Settings**) with script statements enclosed in brackets []. Multiple statements, up to a maximum of 250 characters, may be programmed in a function key. Usually, a main script would be loaded in the script window, and the programmed key would simply call one of the subroutines in the main script (possibly with arguments).

Running a Script from a Menu or Toolbar

Running a script from a menu item or toolbar button is similar to running a script from a function key. In the **Menu Designer** application, set the **Action** property of the menu or toolbar item to the desired script statements enclosed in brackets [].







Running a Script from the Host Computer

To run a script from the host system, send the private AccuTerm command:

ESC STX P *script* CR

where ***script*** is the text of a script to execute. Each script statement is separated by a **LF** or **EM** control character, and the entire script is terminated with a carriage return. Often, when a main script is loaded by opening a layout file or from the command line, the script command sent by the host is simply the name of a subroutine or macro to call (possibly with arguments).

Debugging a Script

The script window contains menu items and buttons for single stepping, interrupting and resuming script execution, and setting breakpoints and watch expressions. Use **Debug Step Into** () to execute the next statement; **Debug Step Over** () to execute the next statement, subroutine or function call. Select **Run Pause** or click  to interrupt execution; select **Run Start (Resume)** or click  to resume execution. To set or clear a breakpoint, click on the desired script line, then select **Debug Toggle Breakpoint** or click the  button. Lines containing breakpoints are marked with a large dot in the left margin. To add a watch expression, position the cursor on the desired expression, then select **Debug Add Watch** or click the  button. To remove a watch expression, click on the **Watch** tab, select the expression you want to delete, and press the **Del** key.

Controlling AccuTerm with Scripts

From a script's perspective, AccuTerm consists of a set of objects which can be manipulated. The main object, AccuTerm, may be used to control the general settings of AccuTerm - those items which are set in the **General Settings** or **File Transfer Settings** category of the **Settings** dialog box. The **Session** object is the most useful object for controlling AccuTerm. Using the **Session** object, you can change any of the session **Settings**, communicate with the host system, manipulate the screen, etc. The following chapter, **AccuTerm Object Reference**, describes the object structure used to control AccuTerm.

There are several built-in objects which can be referenced while executing scripts. These are the AccuTerm object, the Sessions collection, the **ActiveSession** object, and the **InitSession** object.

AccuTerm object

The AccuTerm object is the top-level application object, and is described in detail in the next chapter.

Sessions collection

The Sessions collection is a collection Session objects. There is one Session object in the collection for each currently open session. The first session in the collection is `Sessions(0)`. `Sessions` is simply a shortcut for `AccuTerm.Sessions`. The Session object type is described in detail in the next chapter.

ActiveSession object

The ActiveSession object is an object of type Session, which refers to the currently active session. The currently active session is the session whose title bar is highlighted.

InitSession object

The InitSession object is an object of type Session, which refers to the session which initiated execution of the current script. This may not be the same as ActiveSession, since script execution could be initiated from a non-active session under host program control.

Script Language Extensions

Several new statements have been added to the VBA language used by AccuTerm. These extensions implement features which may be useful in AccuTerm scripts.

AppActivate statement

AppActivate **title\$**

Activates the application identified by **title\$**. If no window exists with **title\$**, the first window with a title that begins with **title\$** is activated. If no window matches, an error occurs.

AppClose statement

AppClose [**title\$**]

Closes the named application. The **title\$** parameter is a String containing the name of the application. If the **title\$** parameter is absent, then the AppClose statement closes the active application.

AppFind function

AppFind(**title\$**)

Returns a String containing the full name of the application matching the partial **title\$**.

AppGetActive function

AppGetActive()

Returns a String containing the name of the active application.

AppGetPosition statement

AppGetPosition *X*, *Y*, *width*, *height* [, *title\$*]

Retrieves the position of the named application. *X*, *Y*, *width*, *height* are integer variables into which the application's position and size are stored. ***Title\$*** is the name of the application whose size is being retrieved. If ***title\$*** is not specified, then the currently active application is assumed.

AppGetState function

AppGetState ([*title\$*])

Returns an Integer specifying the state of the top-level window.

AppHide statement

AppHide [*title\$*]

Hides the named application.

AppList statement

AppList ***AppNames\$***()

Fills an array with the names of all open applications. The ***AppNames\$*** parameter must specify either a zero- or one-dimensioned dynamic String array. The array will be redimensioned to match the number of open applications. After calling this function, you can use LBound and UBound to determine the new size of the array.

AppMaximize statement

AppMaximize [*title\$*]

Maximizes the named application.

AppMinimize statement

AppMinimize [*title\$*]

Minimizes the named application.

AppMove statement

AppMove *X*, *Y* [, *title\$*]

Sets the upper left corner of the named application to a given location.

AppRestore statement

AppRestore [*title\$*]

Restores the named application.

AppSetState statement

AppSetState **newstate** [, **title\$**]

Maximizes, minimizes, or restores the named application, depending on the value of **newstate**. The application is maximized if **newstate** is 1, minimized if **newstate** is 2 and restored if **newstate** is 3.

AppShow statement

AppShow [**title\$**]

Shows the named application.

AppSize statement

AppSize **width**, **height** [, **title\$**]

Sets the size of the named application.

Chain statement

Chain **filename** [**arguments**]

Chain **filename** | **macroname** [**arguments**]

This statement transfers control to the specified script file. **Filename** is a string expression, and may contain a complete path name. If **macroname** is present, it must be separated from **filename** by a vertical bar (|) and execution begins with subroutine **macroname**. Otherwise, execution begins with Sub Main() in the new script file. Optional **arguments** may be appended to the filename string separated by a space character. The arguments may be retrieved in the chained-to script by using the Command() function. Execution never returns to the calling script. *Note: both **filename** and **macroname** and optional **arguments** are contained in a single string expression, such as*

“C:\ATWIN\SCRIPTS\MYSCRIPT.SCR|FOO IMAGE XX.JPG”

Common collection

Common.Clear

Common(**key**) = **value**

Variable = Common(**key**)

The Common collection is used to store values which may be saved between script executions, or shared with between several running scripts. That is, one session can set a value in the Common collection, which another session can retrieve it. Items stored in the Common collection are referenced by “keys”. The **key** is a string argument. The values stored in the Common collection are *variants*, and thus can contain any data type. The Common collection can be reset by using the Clear method.

Debug.Print statement

Debug.Print **text**

This statement prints the specified **text** to a special debugging window. Use **Window Debug** menu to show the debugging window.

FileExists function

FileExists(**name\$**)

Returns True if file exists, otherwise returns False.

Item function

Item(**text\$, first, last**, [**delim\$**])

Returns all the items between **first** and **last** within the specified formatted text list. Items are substrings of a delimited text string. Items, by default, are separated by commas and/or end-of-lines. This can be changed by specifying different delimiters in the **delim\$** parameter.

ItemCount function

ItemCount(**text\$, [delim\$]**)

Returns an Integer containing the number of items in the specified delimited text. Items are substrings of a delimited text string. Items, by default, are separated by commas and/or end-of-lines. This can be changed by specifying different delimiters in the **delim\$** parameter.

Line function

Line(**text\$, first, last**)

Returns a String containing a single line or a group of lines between **first** and **last**.

LineCount function

LineCount(**text\$**)

Returns an Integer representing the number of lines in **text\$**.

OpenFileName function

OpenFileName([**title\$** [, **extension\$**]])

Displays a dialog box that prompts the user to select from a list of files, returning the full pathname of the file the user selects or a zero-length string if the user selects Cancel. If **title\$** is specified, then it is used as the title of the dialog box, otherwise, "Open" is used. If **extension\$** is specified, then it specifies a list of available file types.

Pause statement

Pause **Seconds**

This statement causes the script to pause for the specified number of seconds. While the script is paused, normal terminal functions are operational, including any file transfers in progress.

Random function

Random(**min**, **max**)

Returns a Long value greater than or equal to **min** and less than or equal to **max**.

SaveFileName function

SaveFileName([**title\$** [, **extension\$**]])

Displays a dialog box that prompts the user to select from a list of files, returning the full pathname of the file the user selects or a zero-length string if the user selects Cancel. If **title\$** is specified, then it is used as the title of the dialog box, otherwise, "Save As" is used. If **extension\$** is specified, then it specifies a list of available file types.

Sleep statement

Sleep **Milliseconds**

This statement causes the script to pause for the specified number of milliseconds. While the script is paused, normal terminal functions are operational, including any file transfers in progress.

Word function

Word(**text\$**, **first**, **last**)

Returns a String containing a single word or a group of words between **first** and **last**.

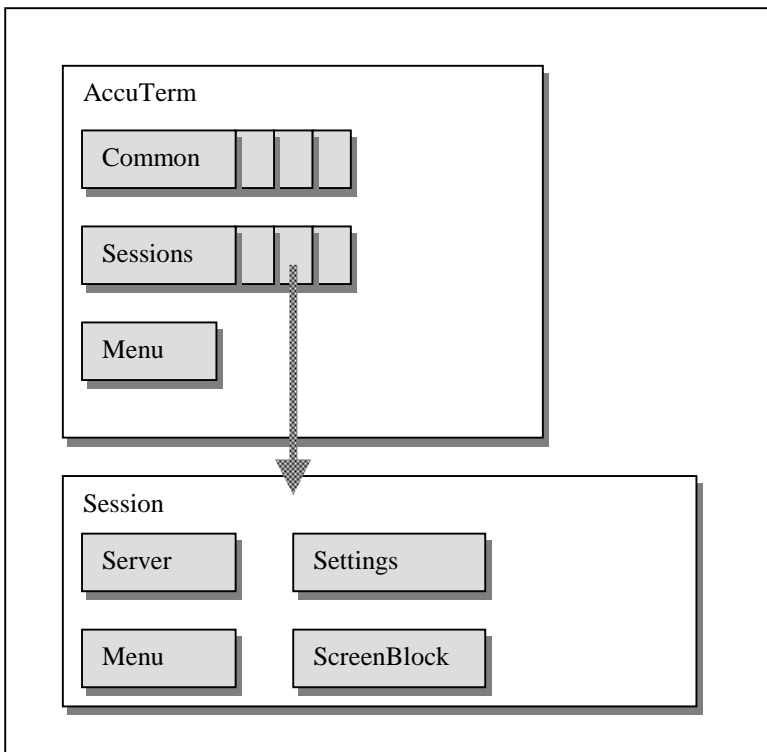
WordCount function

WordCount(**text\$**)

Returns an Integer representing the number of words in **text\$**. Words are separated by spaces, tabs, and end-of-lines.

ACCUTERM OBJECT REFERENCE

AccuTerm 97 exposes a rich OLE/ActiveX object structure. This allows AccuTerm to act as an automation server to any number of client applications. AccuTerm's objects are arranged in a hierarchy, with the AccuTerm application object at the top. The object hierarchy is shown in diagram below.



Properties

This following sections describe the properties, methods and events available from AccuTerm' objects. To reference a property, simply append a period (.) followed by the property name to an object variable of the appropriate type. To get the value of a property, simply assign the property reference to a variable of the appropriate type; to set the value, assign an appropriate expression, constant or variable to the property reference.

Methods

To reference a method, simply append a period (.) followed by the method name to an object variable of the appropriate type. Some methods return a value; for these methods, assign the result to a variable of the appropriate type. Some methods accept one or more arguments separated by commas. If the method also returns a value, enclose the argument list in parentheses. Optional arguments are shown enclosed in brackets []; the brackets are not part of the syntax.

Events

Certain of AccuTerm's objects have the ability to fire events. To use events, the client application must be capable of responding to events. For example, to use the `Session` object's `DataReady` event, the following declaration is required in a VB form or class:

```
Dim WithEvents objSession As  
AccuTermClasses.Session
```

Type Library

AccuTerm 97 includes a type library, `ATWIN32.TLB`, which contains information about AccuTerm's public objects, including their properties, methods, events, arguments and constants. When specifying the object type in a declaration, always use the type library as a reference. This will ensure compatibility with future versions of AccuTerm. For example, to declare a VB object variable as type "AccuTerm", use the following declaration:

```
Dim obj As AccuTermClasses.AccuTerm
```


The AccuTerm Object

The AccuTerm object is AccuTerm's top-level application object. To access the AccuTerm object of a running instance of AccuTerm, use the following syntax:

```
Dim obj as Object
Set obj = GetObject(, "ATWin32.AccuTerm")
```

To create a new instance of AccuTerm, use:

```
Dim obj as Object
Set obj = CreateObject("ATWin32.AccuTerm")
```

The properties and methods of the AccuTerm object are described in the following pages. Use these properties and methods to control the overall application. The AccuTerm object contains the Sessions collection, which has one element for each open session. The ActiveSession property may be used to obtain a reference to the Session object which is currently active.

Activate method

Makes AccuTerm the active application.

ActiveSession property (Session object)

The ActiveSession object is an object of type Session, which refers to the currently active session. The currently active session is the session whose title bar is highlighted.

Arrange method

AccuTerm.Arrange [**style**]

Arranges AccuTerm session windows. Set **style** to 0 for cascade, 1 for tile horizontal, 2 for tile vertical and 3 to arrange icons.

AsciiDelay property (integer)

Specifies the interline delay used for ASCII file uploads in milliseconds.

AsciiEOL property (integer)

Specifies the line ending sequence for ASCII file uploads. Set to 0 for **CR**, 1 for **LF** or 2 for **CR+LF**.

AutoClose property (boolean)

If non-zero, then when the last open session is closed, AccuTerm will terminate.

Close method

`AccuTerm.Close [prompt]`

Closes AccuTerm. If **prompt** is True (non-zero) and any changes have been made to session configuration settings, the user will be prompted save the changed settings.

This is the same as the `Terminate` method, but is *not* available when using late binding.

Common property (collection object)

Returns a reference to the Common collection object. The Common collection may be used to save global variables so that they may be shared between different instances of scripts and OLE clients.

CustomMouseTable property (string)

This is the file name of the current custom mouse table. See **Appendix F** for more information on using a custom mouse table.

DefaultCaptureDir property (string)

This is the default destination directory used for file capture operations which do not specify a directory.

DefaultXferDir property (string)

This is the default destination directory used for file transfer operations which do not specify a directory.

DefaultXferMode property (integer)

This is default transfer mode: 0 for text, 1 for binary.

DefaultXferOverwrite property (integer)

This is default overwrite setting for received files. Set to True (non-zero) to allow overwrites, else set to False (zero).

Height property (long)

This is the application window height in pixels.

Hide method

This method makes AccuTerm's main window invisible.

KermitCRC property (integer)

Set to True (non-zero) to enable CRC block checking, else set to False (zero) to use checksum block checking. If this option is enabled, and the

host Kermit supports this option, then a 16 bit CRC is used to verify packet integrity.

`KermitEOL` property (integer)

Specifies the ASCII character to be sent at the end of each Kermit packet.

Usually 13 (**CR**), but may be 10 (**LF**) for some hosts.

`KermitErrRetry` property (integer)

Specifies the number of times Kermit will retry during a file transfer before the transfer is aborted.

`KermitInitRetry` property (integer)

Specifies the number of times Kermit will retry when initiating a file transfer before the transfer is aborted.

`KermitQuote` property (integer)

Set to True (non-zero) to enable the “eighth bit quoting” option, else set to False (zero). If this option is enabled, and the host Kermit supports this option, then any characters which have the eighth bit set are “quoted” using a sequence of characters which do not have the eighth bit set. This option is useful for connections which do not support eight bit data.

`KermitRept` property (integer)

Set to True (non-zero) to enable the run length encoding data compression option, else set to False (zero). If this option is enabled, and the host Kermit supports this option, then repeated sequences of any character are encoded to compress the data.

`KermitTimeout` property (integer)

Specifies the number of seconds before Kermit file transfer operations time out.

`Left` property (long)

This is the horizontal position of the application window in pixels.

`Menu` property (Menu object)

This returns a reference to the application Menu object, which may be used to customize AccuTerm’s menus and toolbars. For more information on the Menu object, refer to **Appendix F**.

MenuFileName property (string)

This is the name of a custom menu file. Use the menu designer to create custom menus. For more information on the menu designer, refer to **Appendix F**.

Move method

AccuTerm.Move **left** [, **top** [, **width** [, **height**]]]

Repositions and resizes AccuTerm's main window. The positions and dimensions are specified in pixels.

PhoneBookName property (string)

This is the file name of the Auto-Dial phone directory.

ProductLicenseType property (integer)

This read-only property returns the product license type: 1=single user, 2=site license, 3=enterprise license, 9=demonstration version.

ProductName property (string)

This read-only property returns the string "ACCUTERM/WIN32".

ProductRelease property (string)

This read-only property returns the current release of AccuTerm 97.

ProductSerialNumber property (long)

This read-only property returns the serial number of AccuTerm 97.

RegisteredCompany property (string)

This read-only property returns the registered company name.

RegisteredLocation property (string)

This read-only property returns the registered location.

RegisteredUser property (string)

This read-only property returns the registered user name.

Resize method

AccuTerm.Resize **width, height**

Resizes AccuTerm's main window. The dimensions are specified in pixels.

Sessions property (collection)

Returns a reference to the Sessions collection. This collection has one member for each open session. The first session is Sessions(0).

Show method

Makes AccuTerm's main window visible.

SingleInstance property (integer)

Set to True (non-zero) to allow only one instance of AccuTerm. To allow multiple instances, set to False (zero).

StatusLineVisible property (integer)

Set to True (non-zero) to display AccuTerm's status line, otherwise set to False (zero).

TelnetHostsFile property (string)

This is the file name of the hosts file. The hosts file is used to list available hosts when setting up a telnet session.

Terminate method

AccuTerm.Terminate [**prompt**]

Closes AccuTerm. If **prompt** is True (non-zero) and any changes have been made to session configuration settings, the user will be prompted save the changed settings

This is the same as the Close method, and is available when using late binding.

ToolbarVisible property (integer)

Set to True (non-zero) to display AccuTerm's tool bar, otherwise set to False (zero).

Top property (long)

This is the vertical position of the application window in pixels.

Visible property (boolean)

Set to True (non-zero) to make AccuTerm's main window visible. Set to False to hide AccuTerm's main window.

Width property (long)

This is the application window width in pixels.

WindowState property (integer)

This is the state of AccuTerm's main window. Set to 0 for normal, 1 for minimized or 2 for maximized.

XmodemTimeout property (integer)

Specifies the number of seconds before Xmodem file transfer operations time out.

YmodemTimeout property (integer)

Specifies the number of seconds before Ymodem file transfer operations time out.

ZmodemTimeout property (integer)

Specifies the number of seconds before Zmodem file transfer operations time out.

ZmodemAuto property (integer)

Set to True (non-zero) to enable automatic Zmodem downloads, otherwise set to False (zero).

The Sessions Collection

You can create (open) a new session, and you can access any opened session by using the `Sessions` collection. The `Sessions` collection acts like an array of `Session` objects with one element for each open session.

The `Sessions` collection is used to access any open session by using an array reference:

`Sessions(index)`

returns a `Session` object for session *index*. Session indexes are numbered from zero; the first open session is `Sessions(0)`. The last session is `Sessions(Sessions.Count() - 1)`.

Sessions.Add method

```
set object = Sessions.Add ([filename [, state [,  
                             mode [, hidden]]]])
```

The Sessions.Add method creates and initializes a new session. The return value is a Session object: All arguments are optional:

filename	configuration file name used to initialize session.
state	initial window state (0 = normal, 1 = minimized, 2 = maximized)
mode	input mode (0 = normal, 1 = synchronous, 2 = disconnected)
hidden	0 if new session is initially visible, 1 if hidden
object	session object created by this method

Sessions.Count method

```
count = Sessions.Count ( )
```

The Sessions.Count method returns the number of open sessions.

The Session Object

When you use AccuTerm, you create “sessions” to communicate with host computer systems. AccuTerm’s multiple document interface (MDI) allows you to create as many sessions as you desire, all connected to different host systems (or more than one session to a single host). AccuTerm’s Session object gives you access to session properties such as port, font, screen colors, etc.) and allows you to invoke session methods (create, delete, input, output, etc.)

Predefined Session Objects

The currently active session may be accessed using the ActiveSession object. When a script is invoked from a function key, menu, toolbar button, the host computer system, or from DDE, you can access the session which invoked the script by referencing the InitSession object. *The InitSession object is only valid in the context of a script running in the AccuTerm application.*

Creating a New Session

There are two ways to create a new session. First, you can declare a `Session` object variable, and use the `New` keyword with the `Set` statement to create the new session:

```
Dim S as Session
Set S = New Session
```

These statements will create a new session. All session properties will be set to their default values. You access the properties and methods of the session using the `Session` object variable `S`.

An alternate way of creating a new session is to use the `Add` method of the `Sessions` collection object as described above.

Session Object Reference

`Activate` method

Makes the session the “active” session. There is no return value.

`Answerback` property (string)

This is the string that `AccuTerm` returns to the host system when the host requests the “answerback message”.

`AutoAnswer` property (boolean)

If this setting is non-zero, `AccuTerm` will answer incoming calls when the session is connected to a modem.

`AutoClose` property (boolean)

If this setting is non-zero, `AccuTerm` will close the session when it becomes “disconnected”.(modem disconnects or host system drops network connection).

`Baud` property (integer)

This is the baud rate used by the serial port attached to the session. This property is only meaningful when the communication device type is “Serial Port”. Acceptable baud rates are `atBaud300`, `atBaud1200`, `atBaud2400`, `atBaud9600`, `atBaud14400`, `atBaud19200`, `atBaud38400`, `atBaud57600` and `atBaud115200`.

BoldFont property (boolean)

Set to True (non-zero) if terminal font is bold, False (zero) if font is normal.

Break method

Sends a “break” signal to the host connected to the session. There is no return value.

BytesIn property (long)

This read-only property returns the number of bytes received from the host since the session was opened.

BytesOut property (long)

This read-only property returns the number of bytes transmitted to the host since the session was opened.

Caption property (string)

Set or returns the session caption (title).

Capture method

session.Capture *filename* , *source* , *mode*

Initiates or terminates data capture mode. If **source** is zero, capture mode for session is terminated. Set **source** to 1 to initiate capture of received data, 2 to initiate capture of printed data. Set **mode** to 0 to create a new file, 1 to overwrite an existing file, 2 to append to existing file, or 3 to capture to the clipboard instead of a file. If **source** is zero or **mode** is 3, then **filename** is ignored, otherwise it is the name of the file where captured data is saved.

CaptureEnd method

Terminates any capture operation and closes the capture file.

CaptureFileName property (string)

The name of the file name used in the last capture operation. This property is read-only.

CaptureMode property (integer)

The current capture mode (0 to create a new file, 1 to overwrite an existing file, 2 to append to existing file, or 3 to capture to the clipboard instead of a file). This property is read-only.

CaptureSource property (integer)

The current capture source (1 to capture received data, 2 to capture printed data). This property is read-only.

Changed property (integer)

Non-zero if the settings for the session have been modified. Changing this property to zero prevents the warning about saving changes when the session is closed.

Clear method

session.Clear [*left* [, *top* [, *right* [, *bottom* [, *color*]]]]]

Clears block of text from session screen to specified background color.

Default coordinates are the upper left and lower right corners of the screen.

To specify **color**, use the color index described under Color property.

ClearSelection method

This removes any selection rectangle from the session screen. There is no return value.

This method is the same as the Deselect method.

Close method

session.Close [*prompt*]

This closes the session. If **prompt** = 1 or 3, and if any session settings have been modified subsequent to loading or saving the session, the user will be prompted whether to save the settings. If **prompt** = 2 or 3, and the session is connected via a network or dialup connection, the user will be prompted to disconnect.

This is the same as the Terminate method, but is not available when using late binding.

Col , Row properties (integer)

The current cursor column or row. The leftmost column is zero and the topmost is zero. Also see the Locate method.

Color property (integer)

This is the foreground/background color index used when text is displayed using the `SetText` method. To determine the color index, select the foreground and background colors from the following table, and add the indices of each to form the correct color index:

<u>Color</u>	<u>Foreground</u>	<u>Background</u>
Black	0	0
Blue	1	16
Green	2	32
Cyan	3	48
Red	4	64
Magenta	5	80
Brown	6	96
Light Grey	7	112
Dark Grey	8	128
Light Blue	9	144
Light Green	10	160
Light Cyan	11	176
Light Red	12	192
Light Magenta	13	208
Yellow	14	224
White	15	240

Note: the actual colors may be different if the palette has been modified!

`Colors()` property (array of integer)

This array maps foreground and background colors (using the color index described above) to visual attribute combinations. The index to the `Colors()` array is the attribute code shown in the following table:

<u>Index</u>	<u>Attribute Description</u>	<u>Index</u>	<u>Attribute Description</u>
0	normal	22	dim reverse blink
1	blank	24	dim underline
2	blink	26	dim underline blink
4	reverse	28	dim underline reverse
5	reverse blank	30	dim underline reverse blink
6	reverse blink	32	bright
8	underline	34	bright blink
10	underline blink	36	bright reverse
12	underline reverse	38	bright reverse blink
14	underline reverse blink	40	bright underline
16	dim	42	bright underline blink
18	dim blink	44	bright underline reverse
20	dim reverse	46	bright underline reverse blink

`Cols`, `Rows` properties (integer)

The current number of columns or rows. These are read-only properties. Changing the screen mode between normal or extended mode (see `ExtMode`, `NormMode` and `ScrMode` methods), or changing the value of the `ExtCols`, `ExtRows`, `NormCols` or `NormRows` properties may affect the `Cols` and `Rows` properties.

`Connected` property (boolean)

This read-only property reflects the connection status of the session. For serial connections, it reflects the state of the CD signal; for network connections, it is non-zero if the network connection is OK.

`ConnectTimeout` property (integer)

This is the number of seconds to wait while attempting to connect before a timeout error occurs.

Copy method

session.Copy [*left* [, *top* [, *right* [, *bottom*]]]]

Copies block of text from session screen to the clipboard. Default coordinates are the upper left and lower right corners of the screen.

CopyHistory method

session.CopyHistory [*left* [, *top* [, *right* [, *bottom*]]]]

Copies block of text from session history buffer to the clipboard. Default coordinates are the upper left and lower right corners of the history buffer.

CursorType property (boolean)

Set to True (non-zero) if cursor shown as a block, False (zero) if cursor shown as an underline.

DataBits property (integer)

This is the number of data bits used by the serial port attached to the session. This property is only meaningful when the communication device type is "Serial Port". Acceptable values are 7 or 8. After changing this property, use the `Reset` method for the change to take effect.

DataReady event

object.DataReady()

This event is fired when the session's `InputMode` property is non-zero, and data has been received from the host and is ready to be read using the `Input` method.

Delete method

This deletes (closes) the session. There is no return value.

Deselect method

This removes any selection rectangle from the session screen. There is no return value.

This method is the same as the `ClearSelection` method.

Device property (integer)

This is the communications device type attached to the session. The device type may be `atDevNONE` for no device (disconnects session), `atDevSERIAL`, `atDevPICLAN`, `atDevTELNET` or `atDevNASI`.

Dial method

result = **session**.Dial(**PhoneNumber**)

Uses the Auto-Dialer to dial a specified phone number. The return value specifies the result of the dialing operation. See the DialStatus property for a description of the return codes.

DialStatus property (integer)

This read-only property reflects the status of the last Dial or HangUp method. Status codes are described in the following table:

<u>Status</u>	<u>Description</u>
0	Not connected (successful hang-up).
1	Connected (successful dial).
2	Unable to initialize modem.
3	Requested number is busy.
4	Requested number did not answer.
5	Unable to connect.
6	Unable to hang-up.
7	Modem is in use.
8	Invalid phone number.
9	Billing rejected.
10	No dial tone.

Download method

result = **session**.Download(**target** , **protocol** ,
binary [, **overwrite**])

Downloads a file from a host system to the user's PC. **Target** is the name of the destination file (ASCII or XModem protocol), or the name of the destination directory (Kermit, YModem or ZModem protocol). **Protocol** is atProtocolASCII, atProtocolKermit, atProtocolXmodem, atProtocolYmodem or atProtocolZmodem. **Binary** is True (non-zero) for binary transfer mode, False (zero) for text transfer mode. **Overwrite** must be atProtect or atOverwrite, and is only meaningful for ASCII and Xmodem downloads.

Duplex property (integer)

This property sets the communications duplex mode to local (no communication to or from host: atDuplexLOCAL), full (remote echo: atDuplexFULL) or half (local echo: atDuplexHALF).

Emulate method

session.Emulate ***text***

Causes the terminal emulator to process the specified text, including control codes and escape sequences.

ExtCols, ExtRows properties (integer)

These properties define the terminal screen dimensions when the terminal is placed into “extended mode”, also known as “132 column mode”.

Extension event

object_Extension(***text*** as String)

This event is fired when one of the extension prefix and suffix characters have been received by the emulation engine. This event is used to extend the functionality of the emulation engine. Use the SetExtension method to specify the prefix and suffix characters.

ExtMode method

Causes the session to switch to extended (132 column) mode.

FileName property (string)

This is the name of the configuration file opened by the session. If there is no configuration file, then this property is NULL. If you assign a new name to this property, the session will be re-initialized with the settings from the new file.

FKeys () property (array of string)

This array contains the programmed function and editing key strings. The index is formed by combining (adding) the modifier value with the virtual key number. The string value of the programmed key is binary, and might not contain printable characters.

<u>Modifier</u>	<u>Shift</u>	<u>Ctrl</u>	<u>Alt</u>	<u>Modifier</u>	<u>Shift</u>	<u>Ctrl</u>	<u>Alt</u>
0	no	no	no	4000	no	no	yes
1000	yes	no	no	5000	yes	no	yes
2000	no	yes	no	6000	no	yes	yes
3000	yes	yes	no	7000	yes	yes	yes

<u>Virtual Key</u>	<u>Number</u>	<u>Virtual Key</u>	<u>Number</u>
F1	112	Backspace	8
F2	113	Tab	9
F3	114	Insert	45
F4	115	Delete	46
F5	116	Home	36
F6	117	End	35
F7	118	Page Up	33
F8	119	Page Down	34
F9	120	Left	37
F10	121	Right	39
F11	122	Up	38
F12	123	Down	40
F13	124	Escape	27
F14	125	Enter	13
F15	126	Keypad Enter	253
F16	127		

FontName property (string)

This is the name of the terminal font used by the session.

FontSize property (integer)

This is the size of the terminal font in points.

GetBlock method

scrblk = **session**.GetBlock (*left* , *top* , *right* ,
bottom , [*page*])

Creates a ScreenBlock object containing the contents of the specified block. The returned object contains all text, attribute, color, protect and character set information contained within the block. Default **page** is the

current page. This method, when used with the `SetBlock` method, may be used to effect “windowing”. The return value of this method must be assigned to a variable which has been declared as type `ScreenBlock`.

`GetText` method

```
text = session.GetText ([col [, row [, cols [,  
                        NoProtect ]]])
```

Returns the text contents of a row on the session screen. Default coordinate is the current cursor location. Default number of columns is to the end of the row. If **NoProtect** is True (non-zero), only un-protected characters are returned (protected characters are returned as blanks), otherwise, all ANSI characters are returned.

`GmodeEnable` property (boolean)

Set to True (non-zero) if Tektronix graphics mode is enabled for the session; otherwise set to False (zero).

`Handshake` property (integer)

This is the handshake (flow control) method used by the serial port attached to the session. This property is only meaningful when the communications device type is `atDevSERIAL`. Acceptable handshake settings are `atHandshakeNONE`, `atHandshakeXON`, `atHandshakeRTS` or `atHandshakeDTR`.

`Hangup` method

```
result = session.Hangup ( )
```

Uses the Auto-Dialer to disconnect (hangup) a modem call. The return value specifies the result of the hangup operation. See the `DialStatus` property for a description of the return codes.

`Height` property (long)

This is the session window height in pixels.

`Hide` method

This method makes the session window invisible.

`HistoryRows` property (integer)

This is the number of rows allocated for the session history buffer. If a buffer is allocated, each time a line is scrolled off the screen, the line is placed in the history buffer. Likewise, whenever the screen is cleared, the contents of the screen before clearing is copied to the history buffer.

HostName property (string)

This is the name of the host computer system which a session is connected to. This property is only meaningful when the communications device type is “PicLan” or “Telnet”. In the case of Telnet connections, you can use the host IP address in place of the host name. After changing this property, use the `Reset` method for the change to take effect.

HostPort property (string)

For Telnet connections, this is the host TCP/IP port number assigned for Telnet services. For PicLan connections, it is the Pick pseudo-port number; when -1 (negative one) is used, it means “any available port”.

HostTermType property (string)

For Telnet connections, this is the terminal type sent to the host when initiating the connection. This setting is only needed if AccuTerm’s default terminal type is not compatible with the host terminal names.

hWnd property (long)

This read-only value is the window handle for the session’s terminal screen. This handle may be used in Windows API calls in conjunction with the `Declare` statement.

Icon property (string)

This is the name of the icon file used for the session.

ID property (string)

This is the unique session ID.

Input method

```
result = session.Input([mode [, maxlen [, timeout]])
```

This method accepts input from the communications device attached to the session. If **mode** is 0 (default), raw data is returned, up to **maxlen** characters or until **timeout** seconds have passed. If **mode** is 1, a “line” of data is returned, up to **maxlen** characters or until **timeout** seconds have passed. A “line” is terminated by a **CR**, **LF**, **CR+LF** or **LF+CR**. The terminator is not returned as part of the input. Default values for **maxlen** and **timeout** are 80 and 30 respectively.

This method is the same as the `ReadText` method, but is not available when using late binding.

InputMode property (integer)

This specifies how received data is processed by the terminal session. If the InputMode property is set to 0 (normal), any received data is immediately processed by the terminal emulator. Use this mode when the script is not concerned with the data stream. Setting the mode to 1 (synchronous) allows the terminal emulator to process all data which the script has processed. That is, all received data examined by a WaitFor method or Input method is also processed by the emulator. Finally, setting the mode to 2 (disconnected) disconnects the terminal emulator from the received data stream. It is up to the script to pass any data to the emulator using the Emulate method. If a session was created with the Sessions.Add method, the initial InputMode setting may be specified as an argument.

Left property (long)

This is the horizontal position of the session window relative to the upper-left corner of the inside of AccuTerm's main window. This value is in pixels.

LegibleFont property (boolean)

Set to True (non-zero) to keep fonts legible when scaling fonts to fit screen; otherwise set to False (zero).

LoadImage method

```
session.LoadImage filename , col , row [ , width  
[ , height [ , preserveaspect  
[ , borderstyle] ] ] ]
```

Creates an image on the terminal screen at the location specified by **col** and **row**. **Filename** specifies the name of the image file. Image file formats supported are bitmap, Windows metafile, TIFF, TARGA and JPEG. If **width** and **height** are specified, then the image is scaled to fit the specified area; otherwise the image is displayed in its actual dimensions. If **preserveaspect** is non-zero, then the aspect ratio of the original image is preserved by reducing either width or height. To enclose the image in a border, specify **borderstyle** of 0 for none, 1 for simple, 2 for raised or 3 for inset. Multiple images may be loaded on the terminal screen. Images are considered “protected” data; a “clear un-protected characters” command will not clear images from the terminal screen.

Locate method

session.Locate *col* , *row* [, *page*]

Moves the cursor to the specified screen location and optionally changes pages.

LockHistory property (boolean)

Set to True (non-zero) to prevent the history buffer from being updated; otherwise set to False (zero).

LockBaudRate property (boolean)

Set to True (non-zero) to lock the modem to the specified baud rate; otherwise set to False (zero). Note: not all modems support this setting.

LockKeyboard property (boolean)

Set to True (non-zero) to prevent keystrokes from being sent to the host system; otherwise set to False (zero).

MapUpperFKeys property (boolean)

Setting this property to True (non-zero) causes AccuTerm to map **Ctrl+F1** through **Ctrl+F10** as **F11** to **F20**. This is the default behavior, since the PC keyboard does not have keys **F13** to **F20**.

Menu property (Menu object)

This returns a reference to the session Menu object, which may be used to customize AccuTerm's menus and toolbars. For more information on the Menu object, refer to **Appendix F**.

MenuFileName property (string)

This is the name of a custom menu file. Use the menu designer to create custom menus. For more information on the menu designer, refer to **Appendix F**.

MouseEnable property (boolean)

Set to True (non-zero) to enable the ESC STX "I" AccuTerm mouse-on command; otherwise set to False (zero).

MouseTableAdd method

session.MouseTableAdd *button* [, *pattern* [, *click* [, *dblclk*]]]

Adds an entry to the mouse pattern table for the session. The mouse pattern table is used to associate patterns on the screen with responses to be sent to the host when the mouse is clicked. **Button** is 1 for left button, 2 for right

button and 3 for middle button. **Pattern** is a string containing a “regular expression” describing the pattern to be matched. **Click** is a string, and is sent to the host when the specified button is clicked over the pattern. **Dbclick** is the response if the button is double-clicked over the pattern. See Appendix F for details on using the mouse pattern table feature.

MouseTableLoad method

session.MouseTableLoad **filename**

Loads a mouse pattern table from a file. See Appendix F for the format of the mouse pattern table file.

MouseTableReset method

Clears the mouse pattern table.

Move method

session.Move **left** [, **top** [, **width** [, **height**]]]

Repositions and resizes the session window. The positions and dimensions are specified in pixels, and are relative to the upper-left corner of the inside of AccuTerm’s main window.

NormCols, NormRows properties (integer)

These properties define the terminal screen dimensions when the terminal is placed into “normal mode”, also known as “80 column mode”.

NormMode method

Causes the session to switch to normal (80 column) mode.

Output method

session.Output **expr**

This method transmits data to the host using the communications device attached to the session. Transmits **expr** (a string expression) to the connected host computer.

This method is the same as the WriteText method.

OverrideModemConfig property (boolean)

Set to True (non-zero) to override modem Control Panel settings for baud rate, data bits, stop bits, parity and handshake with current session settings for these properties; otherwise set to False (zero).

Page property (integer)

This specifies the number of the current video page for the session. The value must be between 0 and the number of pages allocated for the session minus 1 (see Pages property).

Pages property (integer)

This specifies the number of video pages for the session. The value must be between 1 and 25.

Palette() property (array of OLE_COLOR)

This array contains the color values for the 16 palette entries for the session. The default color assignment for each palette entry is shown in the table for the Color property as “foreground color.” Use the RGB() function to produce an OLE_COLOR.

Parity property (integer)

This is the parity setting used by the serial port attached to the session. This property is only meaningful when the communication device type is “Serial Port”. Acceptable values are atParityNONE, atParityEVEN, atParityODD, atParityMARK and atParitySPACE. After changing this property, use the Reset method for the change to take effect.

Paste method

session.Paste [**filename**]

Copies the text contents of the clipboard (or file if **filename** is specified) to the host.

PasteEOFChar property (integer)

This is the ASCII character which is transmitted to the host at the end of paste operation. If this value is zero, no character will be transmitted.

PasteEOFMode property (integer)

Set to 0 if no character is transmitted at the end of a paste operation. Set to 1 if a **SUB** control code is transmitted. Set to 2 to use the character defined by the PasteEOFChar property.

PasteEOLChar property (integer)

This is the ASCII character which is transmitted to the host at the end of each line in a paste operation. If this value is zero, no character will be transmitted.

PasteEOLMode property (integer)

Set to 0 if a **CR** is transmitted at the end of each line of a paste operation. Set to 1 if a **LF** is transmitted. Set to 2 if a **CR+LF** is transmitted. Set to 3 if a **TAB** is transmitted. Set to 4 if no character is transmitted. Set to 5 to use the character defined by the PasteEOLChar property.

PlayMidi method

session.PlayMidi *filename*

Plays the MIDI sound file specified by *filename*.

PlayWave method

session.PlayWave *filename*

Plays the Wave sound file specified by *filename*.

Port property (integer)

Specifies the Com port number attached to the session. This property is only meaningful when the communication device type is atDevSERIAL.

Acceptable values are from 1 to 8. After changing this property, use the Reset method for the change to take effect.

PrinterClose method

session.PrinterClose

Closes the current print job.

PrinterColorMode property (integer)

If this property is non-zero and the ScreenPrintMode property is non-zero (Graphics mode), the screen is printed in color; otherwise the screen is printed in black and white. This property has no effect when the ScreenPrintMode property is zero (Text mode).

PrinterFontBold property (boolean)

Set this property to True (non-zero) to print using boldface. This property affects slave (or Aux) print jobs when the SlavePrintMode property is non-zero (Graphics mode). This property has no effect when the SlavePrintMode property is zero (Text mode).

PrinterFontItalic property (boolean)

Set this property to True (non-zero) to print using italics. This property affects slave (or Aux) print jobs when the SlavePrintMode property is non-zero (Graphics mode). This property has no effect when the SlavePrintMode property is zero (Text mode).

PrinterFontName property (string)

Set this property to the name of the font to use for slave (or Aux) print jobs when the SlavePrintMode property is non-zero (Graphics mode). This property has no effect when the SlavePrintMode property is zero (Text mode).

PrinterFontSize property (integer)

Set this property to the size (in points) of the font to use for slave (or Aux) print jobs when the SlavePrintMode property is non-zero (Graphics mode). This property has no effect when the SlavePrintMode property is zero (Text mode).

PrinterMode property (integer)

This is the state of the auxiliary printer: 0 if printer is off, 1 if printer is auto-print mode, 2 if the printer is in transparent print mode.

PrinterName property (string)

This is the name of the printer used by the session for auxiliary port output. It may be set to null ("") to indicate “no printer”.

PrinterOff method

session.PrinterOff

Turns off the auxiliary printer.

PrinterOn method

session.PrinterOn [*mode*]

Turns on the auxiliary printer. If *mode* is zero (default), printer is in “copy” mode (data is displayed on both the screen and printer). Otherwise, printer is in “transparent” mode (data is sent to the printer only, not the screen).

PrinterOrientation property (integer)

Set this property to 0 use the default page orientation, 1 for portrait or 2 for landscape. This property applies only to slave (or Aux) print jobs and only when the SlavePrintMode property is non-zero (Graphics mode). This property has no effect when the SlavePrintMode property is zero (Text mode).

PrinterPaperSize property (integer)

Set this property to 0 use the default paper size for the printer. Otherwise use a paper size from the table below. This property applies only to slave (or Aux) print jobs and only when the SlavePrintMode property is non-zero (Graphics mode). This property has no effect when the SlavePrintMode property is zero (Text mode).

1 Letter 8 1/2 x 11 in	2 Letter Small 8 1/2 x 11 in
3 Tabloid 11 x 17 in	4 Ledger 17 x 11 in
5 Legal 8 1/2 x 14 in	6 Statement 5 1/2 x 8 1/2 in
7 Executive 7 1/4 x 10 1/2 in	8 A3 297 x 420 mm
9 A4 210 x 297 mm	10 A4 Small 210 x 297 mm
11 A5 148 x 210 mm	12 B4 (JIS) 250 x 354
13 B5 (JIS) 182 x 257 mm	14 Folio 8 1/2 x 13 in
15 Quarto 215 x 275 mm	16 10x14 in
17 11x17 in	18 Note 8 1/2 x 11 in
19 Envelope #9 3 7/8 x 8 7/8	20 Envelope #10 4 1/8 x 9 1/2
21 Envelope #11 4 1/2 x 10 3/8	22 Envelope #12 4 1/2 x 11
23 Envelope #14 5 x 11 1/2	24 C size sheet
25 D size sheet	26 E size sheet
27 Envelope DL 110 x 220mm	28 Envelope C5 162 x 229 mm
29 Envelope C3 324 x 458 mm	30 Envelope C4 229 x 324 mm
31 Envelope C6 114 x 162 mm	32 Envelope C65 114 x 229 mm
33 Envelope B4 250 x 353 mm	34 Envelope B5 176 x 250 mm
35 Envelope B6 176 x 125 mm	36 Envelope 110 x 230 mm
37 Envelope Monarch 3.875x7.5 in	38 6 3/4 Envelope 3 5/8 x 6 1/2 in
39 US Std Fanfold 14 7/8 x 11 in	40 German Std Fanfold 8 1/2 x 12 in
41 German Legal Fanfold 8 1/2x13 in	

`PrinterPaperSource` property (integer)

Set this property to 0 use the default paper source for the printer. Otherwise use a paper source from the table below. This property applies only to slave (or Aux) print jobs and only when the `SlavePrintMode` property is non-zero (Graphics mode). This property has no effect when the `SlavePrintMode` property is zero (Text mode).

- 1 Upper tray
- 2 Lower tray
- 3 Middle tray
- 4 Manual feed
- 5 Envelope feeder
- 6 Envelope manual
- 7 Auto
- 8 Tractor

`PrinterTimeout` property (integer)

This property specifies the number of seconds of inactivity before a print job is closed.

`PrintJobEject` property (integer)

This property specifies if a page is ejected before or after a print job. Set this property to 0 for no page ejects, 1 to eject page before print job, 2 to eject page after print job or 3 to eject page before and after print job.

`PrintScreen` method

Prints the text screen to the printer.

`PrintScreenBackground` property (boolean)

Set to True (non-zero) to print the screen background when using the `PrintScreen` command and the `PrintScreenMode` property is non-zero (Graphics mode); otherwise set to False (zero). This property has no effect when the `ScreenPrintMode` property is zero (Text mode).

`PrintScreenEject` property (integer)

This property specifies if a page is ejected before or after a print screen. Set this property to 0 for no page ejects, 1 to eject page before print screen, 2 to eject page after print screen or 3 to eject page before and after print screen.

`ProtectAttr` property (integer)

Visual attribute number assigned to “protected fields” under Wyse/ADDS emulation. See `Colors()` property for a list of attribute numbers.

ReadText method

result = **session**.ReadText ([**mode** [, **maxlen** [, **timeout**]])

This method accepts input from the communications device attached to the session. If **mode** is 0 (default), raw data is returned, up to **maxlen** characters or until **timeout** seconds have passed. If **mode** is 1, a “line” of data is returned, up to **maxlen** characters or until **timeout** seconds have passed. A “line” is terminated by a **CR**, **LF**, **CR+LF** or **LF+CR**. The terminator is not returned as part of the input. Default values for **maxlen** and **timeout** are 80 and 30 respectively.

This method is the same as the Input method, and is available when using late binding.

Reset method

session.Reset [**mode**]

Resets the terminal (**mode** = 1), the communication device (**mode** = 2) or both (default, **mode** = 0). Note: you may need to reset the communication device after changing certain device related properties in order for the changes to take effect.

ResetComm method

session.ResetComm

Resets the communication device. *Note: you may need to reset the communication device after changing certain device related properties in order for the changes to take effect.*

ResetTerm method

Resets the terminal emulator for session (clears screen, resets protect mode, unlocks keyboard, etc.)

Resize method

session.Resize **width**, **height**

Resizes the session window. The dimensions are specified in pixels.

Save method

session.Save [**filename**]

Saves the session configuration. If **filename** is not specified, the original file name is used. If there is no file name for the session, the user will be prompted for a file name.

ScaleFont property (boolean)

Set to True (non-zero) to enable automatic font scaling; otherwise set to False (zero).

ScrAutoFwd property (boolean)

Set to True (non-zero) to enable the “Screen Auto Forward” feature; otherwise set to False (zero).

ScreenPrintMode property (boolean)

Set to False (zero) for text mode, True (non-zero) for graphics mode. When this property is True, the PrinterColorMode and PrintScreenBackground properties affect how the screen is printed.

ScrMode property (integer)

Set to zero for normal (80 column) mode, 1 for extended (132 column) mode.

ScrollMode property (integer)

This controls the appearance of the vertical scroll bar: 0=no scroll bar (scroll bar is only visible if the current number of Rows will not fit in the current window size), 1=always show scroll bar, and 2=automatically show scroll bar when cursor is positioned near right border of window.

Select method

session.Select [*left* [, *top* [, *right* [, *bottom*]]]]

Places a selection rectangle on the session screen. Defaults are upper left corner and lower right corner of screen.

This method is the same as the SetSelection method, but is *not* available when using late binding.

Selection property (string)

Returns the current selection, if any. Lines are separated by **CR LF**. If there is no current selection, returns null (“”).

Server property (Server object)

Returns a reference to a Server object, which can be used for host database access if the host is in “server” mode. See description of Server object.

ServerMode property (boolean)

This read-only property is non-zero if the host connected to the session is in “server” mode.

ServerName property (string)

This read-only property is the name of server host computer.

SetBlock method

session.SetBlock **ScreenBlock** , **left** , **top** ,[**page**]

Copies the contents of ScreenBlock object **ScreenBlock** to the specified location on the screen. The copied block contains all text, attribute, color, protect and character set information. Default **page** is the current page. This method must be used with the GetBlock method, and may be used to effect “windowing”.

SetExtension method

session.SetExtension **leadin** , **terminator**

If **leadin** is not NULL, then this method enables the session Extension event. This event may be used to extend the functionality of AccuTerm’s emulation engine. The **leadin** string specifies one or more “lead-in” character codes, which, when preceded by the **ESC** control code, signal the start of an “extended” function. Receipt of one of the characters in the **terminator** string signals the end of the “extended” function. If **terminator** is NULL, then the next character after one of the **leadin** characters signals the end of the “extended” function. When the Extension event is fired, the string beginning with the **leadin** character and ending with the **terminator** character is passed to the event handler. If leadin is NULL, then the Extension event is disabled. *Note: only characters which form invalid terminal escape sequences should be used for leadin, otherwise the extension will not function.*

SetSelection method

session.SetSelection [**left** [, **top** [, **right** [, **bottom**]]]]

Places a selection rectangle on the session screen. Defaults are upper left corner and lower right corner of screen.

This method is the same as the Select method, but is available when using late binding.

SetText method

session.SetText **text** [, **col** [, **row** [, **color**]]])

Copies **text** (a string expression) to the specified location on the session screen. Default location is the current cursor location. Default color is last color used, or value assigned to the Color property. The cursor location is updated by this method.

Settings method (Settings object)

This object is used to set session settings as a block. It is intended to be used in a With Settings . . . End With structure. When used in this manner, all of the settings which are modified are applied to the session when the End With statement is executed. This prevents errors when certain settings depend on others (such as Port and Baud) and also improves performance. Most of the Session properties may be set using this object.

Show method

Makes the session window visible.

ShowErrs property (boolean)

Set to True (non-zero) to enable notification of communications errors; otherwise set to False (zero).

SlavePrintMode property (boolean)

Set to False (zero) for text mode, True (non-zero) for graphics mode. When this property is True, the PrinterFont..., PrinterOrientation, PrinterPaperSize and PrinterPaperSource properties affect how the slave (Aux) print job is printed.

Sound property (string)

This property may be used to specify a custom sound for the terminal beep. Set this property to null to use the default sound. Set to a valid .wav file name to play the specified wave file. Set to "SystemDefault", "SystemHand", "SystemExclamation" or "SystemAsterisk" to use the sound associated with a system event (as defined in the Control Panel Sounds applet). Set to *frequency,duration* to use a true "beep" at the specified frequency (Hz) and duration (ms).

StopBits property (integer)

This is the number of stop bits used by the serial port attached to the session. This property is only meaningful when the communication device type is “Serial Port”. Acceptable values are 1 or 2. After changing this property, use the `Reset` method for the change to take effect.

Strip8th property (boolean)

Set to True (non-zero) to cause `AccuTerm` to truncate received data to 7 bits; otherwise set to False (zero).

TelnetAltBreak property (boolean)

Set to True (non-zero) to send the Telnet Break command when the **Break** key is pressed; otherwise set to False (zero) for to send the Telnet Interrupt Process command.

TelnetBinary property (boolean)

Set to True (non-zero) to enable Telnet binary communication mode; otherwise set to False (zero) for text communication mode.

TelnetBypass property (boolean)

Set to True (non-zero) to bypass initial option negotiation; otherwise set to False (zero) for normal negotiation. This is required for certain hosts like Pick’s D3 which do not implement the complete Telnet protocol.

TelnetEcho property (boolean)

Set to True (non-zero) to enable Telnet remote-echo mode; otherwise set to False (zero) for local echo mode. Note: when using local echo, you should also set the `Duplex` property to “half”.

Terminate method

session.`Terminate` [*prompt*]

This closes the session. If **prompt** = 1 or 3, and if any session settings have been modified subsequent to loading or saving the session, the user will be prompted whether to save the settings. If **prompt** = 2 or 3, and the session is connected via a network or dialup connection, the user will be prompted to disconnect.

This method is the same as the `Close` method, and is available when using late binding.

TermType property (integer)

This is the terminal emulation setting for the session. Possible values are atTermTTY (basic TTY), atTermVPA2 (ADDS Viewpoint A2), atTermVP60 (ADDS Viewpoint 60), atTermP60 (ProComm Viewpoint 60), atTermA2E (ADDS Viewpoint A2 Enhanced), atTermWY50 (Wyse 50), atTermWY60 (Wyse 60), atTermVT52 (DEC VT-52), atTermVT220 (DEC VT-220), atTermANSI (ANSI BBS), atTermPICKMON (Pick PC Monitor) or atTermTEK (Tektronix 4014).

Top property (long)

This is the vertical position of the session window relative to the upper-left inside corner of AccuTerm's main window. This value is in pixels.

UnloadImage method

session.UnloadImage [*filename*]

Removes an image created with the LoadImage method from the terminal screen. The image to be removed is identified by **filename**. If **filename** is not specified, the all images are removed.

Upload method

result = **session**.Upload(**source** , **protocol** , **binary**
[, **overwrite**])

Uploads a file from the user's PC to the connected host system. **Source** is the name of the source file(s) (Kermit, Ymodem and Zmodem may use wild-card characters in **source**). **Protocol** is atProtocolASCII, atProtocolKermit, atProtocolXmodem, atProtocolYmodem or atProtocolZmodem. **Binary** is True (non-zero) for binary transfer mode, False (zero) for text transfer mode. **Overwrite** is meaningful for Zmodem only and must be atProtect, atOverwrite, atAppend, atNewer, atUpdate or atResume.

Visible property (boolean)

Set to True (non-zero) to make the session window visible. Set to False to hide the session window.

WaitFor method

result = **session**.WaitFor(**mode**, **timeout**, **string1**
[, ... , **string10**])

This method causes AccuTerm to wait for one or more strings to be received from the host system. Returns the index of the string first matched, or zero (0) if no string matched within **timeout** seconds. If **mode** is zero, a case-

sensitive comparison is performed, otherwise the comparison is case-insensitive. Up to ten target strings may be specified.

Width property (long)

This is the session window width in pixels.

WindowState property (integer)

This is the state of the session window. Set to 0 for normal, 1 for minimized or 2 for maximized.

WriteText method

session.WriteText *expr*

This method transmits data to the host using the communications device attached to the session. Transmits *expr* (a string expression) to the connected host computer.

This method is the same as the Output method.

XferBytes property (long integer)

This read-only property is the number of bytes transferred as a result of the last file transfer operation.

XferFiles property (long integer)

This read-only property is the number of files transferred as a result of the last file transfer operation.

XferStatus property (integer)

This read-only property reflects the status of the last Upload or Download method. Status codes are described in the following table:

<u>Status</u>	<u>Description</u>
0	File transfer successful.
1	Invalid source file or invalid destination directory.
2	File transfer aborted by user.
3	Destination file already exists.
4	File transfer timed out.
7	File transfer protocol failure.
8	X/Y/ZModem require 8 data bits.
9	X/YModem require hardware flow control.

The Settings Object

The `Settings` property of the `Session` object returns a reference to a temporary `Settings` object. This temporary object is used to retrieve or modify a number of properties as a group. When changing multiple properties, using the `Settings` object improves efficiency.

Nearly all of the properties of the `Settings` object are identical to properties of the `Session` object or to properties of the `AccuTerm` object. Those properties which are duplicated in the `Session` object are OK to use from a script; other properties and methods of the `Settings` object should *not* be used in a script.

The recommended way to use the `Settings` object is in a
With `Session.Settings` . . . End With construct.

`Answerback` property (string)

This is the string that `AccuTerm` returns to the host system when the host requests the “answerback message”.

`AttributeMask` property (integer)

The `AttributeMask` property indicates which visual effects are to be displayed. Use 10 for blink and underline, 8 for underline, 2 for blink or zero to disable both blinking and underline.

`AutoAnswer` property (boolean)

If this setting is non-zero, `AccuTerm` will answer incoming calls when the session is connected to a modem.

`Baud` property (integer)

This is the baud rate used by the serial port attached to the session. This property is only meaningful when the communication device type is “Serial Port”. Acceptable baud rates are `atBaud300`, `atBaud1200`, `atBaud2400`, `atBaud9600`, `atBaud14400`, `atBaud19200`, `atBaud38400`, `atBaud57600` and `atBaud115200`.

`BoldFont` property (boolean)

Set to True (non-zero) if terminal font is bold, False (zero) if font is normal.

`Colors()` property (array of integer)

This array maps foreground and background colors (using the color index described above) to visual attribute combinations. See `Session Colors` property for more details.

`ConnectTimeout` property (integer)

This is the number of seconds to wait while attempting to connect before a timeout error occurs.

`CursorType` property (boolean)

Set to True (non-zero) if cursor shown as a block, False (zero) if cursor shown as an underline.

`DataBits` property (integer)

This is the number of data bits used by the serial port attached to the session. This property is only meaningful when the communication device type is "Serial Port". Acceptable values are 7 or 8. After changing this property, use the `Reset` method for the change to take effect.

`DefaultCaptureDir` property (string) [†]

This is the default destination directory used for file capture operations which do not specify a directory.

`DefaultXferDir` property (string) [†]

This is the default destination directory used for file transfer operations which do not specify a directory.

`DefaultXferMode` property (integer) [†]

This is default transfer mode: 0 for text, 1 for binary.

`DefaultXferOverwrite` property (integer) [†]

This is default overwrite setting for received files. Set to True (non-zero) to allow overwrites, else set to False (zero).

`Device` property (integer)

This is the communications device type attached to the session. The device type may be `atDevNONE` for no device (disconnects session), `atDevSERIAL`, `atDevPICLAN`, `atDevTELNET` or `atDevNASI`.

[†] Do not use in scripts

Dialog method

settings.Dialog [**Title**, [**InitTab**, [**Tabs**]]]

This method displays a tabbed settings dialog box which will allow the user to modify most settings. You can specify an optional **Title** for the dialog, as well as the initially selected tab number **InitTab** (first tab is 0), and a list of **Tabs** which are displayed. The **Tabs** list is a string with a single letter for each visible tab. Letters which may be used for **Tabs** are:

P = printer
X = file transfer
D = device
T = term type
S = screen
F = fonts
C = colors
K = function keys
M = miscellaneous

Duplex property (integer)

This property sets the communications duplex mode to local (no communication to or from host: `atDuplexLOCAL`), full (remote echo: `atDuplexFULL`) or half (local echo: `atDuplexHALF`).

ExtCols, ExtRows properties (integer)

These properties define the terminal screen dimensions when the terminal is placed into “extended mode”, also known as “132 column mode”.

FKeys property (collection)

Set **settings**.Fkeys = **collection**

Set **collection** = **settings**.Fkeys()

This property gets or sets all of the programmed function keys. The collection contains one item for each programmed key. The collection key is the function key index (see `Session FKeys` property). Each item is a string consisting of the key index followed by a NUL character followed by the key contents (string of characters). The key contents may contain control characters.

FontName property (string)

This is the name of the terminal font used by the session.

FontSize property (integer)

This is the size of the terminal font in points.

GmodeEnable property (boolean)

Set to True (non-zero) if Tektronix graphics mode is enabled for the session; otherwise set to False (zero).

Handshake property (integer)

This is the handshake (flow control) method used by the serial port attached to the session. This property is only meaningful when the communications device type is atDevSERIAL. Acceptable handshake settings are atHandshakeNONE, atHandshakeXON, atHandshakeRTS or atHandshakeDTR.

HistoryRows property (integer)

This is the number of rows allocated for the session history buffer. If a buffer is allocated, each time a line is scrolled off the screen, the line is placed in the history buffer. Likewise, whenever the screen is cleared, the contents of the screen before clearing is copied to the history buffer.

HostName property (string)

This is the name of the host computer system which a session is connected to. This property is only meaningful when the communications device type is “PicLan” or “Telnet”. In the case of Telnet connections, you can use the host IP address in place of the host name. After changing this property, use the Reset method for the change to take effect.

HostPort property (string)

For Telnet connections, this is the host TCP/IP port number assigned for Telnet services. For PicLan connections, it is the Pick pseudo-port number; when -1 (negative one) is used, it means “any available port”.

HostTermType property (string)

For Telnet connections, this is the terminal type sent to the host when initiating the connection. This setting is only needed if AccuTerm’s default terminal type is not compatible with the host terminal names.

Initialize method [†]

This method returns all emulator settings to their default values.

KermitCRC property (integer) [†]

Set to True (non-zero) to enable CRC block checking, else set to False (zero) to use checksum block checking. If this option is enabled, and the

[†] Do not use in scripts

host Kermit supports this option, then a 16 bit CRC is used to verify packet integrity.

KermitEOL property (integer) [†]

Specifies the ASCII character to be sent at the end of each Kermit packet. Usually 13 (**CR**), but may be 10 (**LF**) for some hosts.

KermitErrRetry property (integer) [†]

Specifies the number of times Kermit will retry during a file transfer before the transfer is aborted.

KermitInitRetry property (integer) [†]

Specifies the number of times Kermit will retry when initiating a file transfer before the transfer is aborted.

KermitQuote property (integer) [†]

Set to True (non-zero) to enable the “eighth bit quoting” option, else set to False (zero). If this option is enabled, and the host Kermit supports this option, then any characters which have the eighth bit set are “quoted” using a sequence of characters which do not have the eighth bit set. This option is useful for connections which do not support eight bit data.

KermitRept property (integer) [†]

Set to True (non-zero) to enable the run length encoding data compression option, else set to False (zero). If this option is enabled, and the host Kermit supports this option, then repeated sequences of any character are encoded to compress the data.

KermitTimeout property (integer) [†]

Specifies the number of seconds before Kermit file transfer operations time out.

LegibleFont property (boolean)

Set to True (non-zero) to keep fonts legible when scaling fonts to fit screen; otherwise set to False (zero).

Load method [†]

settings.Load ***filename***

This method loads all of the emulator settings from the specified INI file.

[†] Do not use in scripts

`LockBaudRate` property (boolean)

Set to True (non-zero) to lock the modem to the specified baud rate; otherwise set to False (zero). Note: not all modems support this setting.

`MapUpperFKeys` property (boolean)

Setting this property to True (non-zero) causes AccuTerm to map **Ctrl+F1** through **Ctrl+F10** as **F11** to **F20**. This is the default behavior, since the PC keyboard does not have keys **F13** to **F20**.

`MouseEnable` property (boolean)

Set to True (non-zero) to enable the ESC STX "I" AccuTerm mouse-on command; otherwise set to False (zero).

`NormCols`, `NormRows` properties (integer)

These properties define the terminal screen dimensions when the terminal is placed into "normal mode", also known as "80 column mode".

`OverrideModemConfig` property (boolean)

Set to True (non-zero) to override modem Control Panel settings for baud rate, data bits, stop bits, parity and handshake with current session settings for these properties; otherwise set to False (zero).

`Pages` property (integer)

This specifies the number of video pages for the session. The value must be between 1 and 25.

`Palette()` property (array of `OLE_COLOR`)

This array contains the color values for the 16 palette entries for the session. The default color assignment for each palette entry is shown in the table for the `Color` property as "foreground color." Use the `RGB()` function to produce an `OLE_COLOR`.

`Parity` property (integer)

This is the parity setting used by the serial port attached to the session. This property is only meaningful when the communication device type is "Serial Port". Acceptable values are `atParityNONE`, `atParityEVEN`, `atParityODD`, `atParityMARK` and `atParitySPACE`. After changing this property, use the `Reset` method for the change to take effect.

`PasteEOFChar` property (integer)

This is the ASCII character which is transmitted to the host at the end of paste operaton. If this value is zero, no character will be transmitted.

PasteEOFMode property (integer)

Set to 0 if no character is transmitted at the end of a paste operation. Set to 1 if a **SUB** control code is transmitted. Set to 2 to use the character defined by the PasteEOFChar property.

PasteEOLChar property (integer)

This is the ASCII character which is transmitted to the host at the end of each line in a paste operation. If this value is zero, no character will be transmitted.

PasteEOLMode property (integer)

Set to 0 if a **CR** is transmitted at the end of each line of a paste operation. Set to 1 if a **LF** is transmitted. Set to 2 if a **CR+LF** is transmitted. Set to 3 if a **TAB** is transmitted. Set to 4 if no character is transmitted. Set to 5 to use the character defined by the PasteEOLChar property.

Port property (integer)

Specifies the Com port number attached to the session. This property is only meaningful when the communication device type is atDevSERIAL. Acceptable values are from 1 to 8. After changing this property, use the Reset method for the change to take effect.

PrinterColorMode property (integer)

If this property is non-zero and the ScreenPrintMode property is non-zero (Graphics mode), the screen is printed in color; otherwise the screen is printed in black and white. This property has no effect when the ScreenPrintMode property is zero (Text mode).

PrinterFontBold property (boolean)

Set this property to True (non-zero) to print using boldface. This property affects slave (or Aux) print jobs when the SlavePrintMode property is non-zero (Graphics mode). This property has no effect when the SlavePrintMode property is zero (Text mode).

PrinterFontItalic property (boolean)

Set this property to True (non-zero) to print using italics. This property affects slave (or Aux) print jobs when the SlavePrintMode property is non-zero (Graphics mode). This property has no effect when the SlavePrintMode property is zero (Text mode).

PrinterFontName property (string)

Set this property to the name of the font to use for slave (or Aux) print jobs when the SlavePrintMode property is non-zero (Graphics mode). This property has no effect when the SlavePrintMode property is zero (Text mode).

PrinterFontSize property (integer)

Set this property to the size (in points) of the font to use for slave (or Aux) print jobs when the SlavePrintMode property is non-zero (Graphics mode). This property has no effect when the SlavePrintMode property is zero (Text mode).

PrinterName property (string)

This is the name of the printer used by the session for auxiliary port output. It may be set to null ("") to indicate "no printer".

PrinterOrientation property (integer)

Set this property to 0 use the default page orientation, 1 for portrait or 2 for landscape. This property applies only to slave (or Aux) print jobs and only when the SlavePrintMode property is non-zero (Graphics mode). This property has no effect when the SlavePrintMode property is zero (Text mode).

PrinterPaperSize property (integer)

Set this property to 0 use the default paper size for the printer. See the PrinterPaperSize property in the Session object for a list of paper sizes. This property applies only to slave (or Aux) print jobs and only when the SlavePrintMode property is non-zero (Graphics mode). This property has no effect when the SlavePrintMode property is zero (Text mode).

PrinterPaperSource property (integer)

Set this property to 0 use the default paper source for the printer. See the PrinterPaperSource property in the Session object for a list of paper sources. This property applies only to slave (or Aux) print jobs and only when the SlavePrintMode property is non-zero (Graphics mode). This property has no effect when the SlavePrintMode property is zero (Text mode).

PrinterTimeout property (integer)

This property specifies the number of seconds of inactivity before a print job is closed.

PrintJobEject property (integer)

This property specifies if a page is ejected before or after a print job. Set this property to 0 for no page ejects, 1 to eject page before print job, 2 to eject page after print job or 3 to eject page before and after print job.

PrintScreenBackground property (boolean)

Set to True (non-zero) to print the screen background when using the Print Screen command and the PrintScreenMode property is non-zero (Graphics mode); otherwise set to False (zero). This property has no effect when the ScreenPrintMode property is zero (Text mode).

PrintScreenEject property (integer)

This property specifies if a page is ejected before or after a print screen. Set this property to 0 for no page ejects, 1 to eject page before print screen, 2 to eject page after print screen or 3 to eject page before and after print screen.

ProtectAttr property (integer)

Visual attribute number assigned to “protected fields” under Wyse/ADDS emulation. See Colors() property for a list of attribute numbers.

Save method [†]

settings.Save filename

This method saves all of the emulator settings in the specified INI file.

ScaleFont property (boolean)

Set to True (non-zero) to enable automatic font scaling; otherwise set to False (zero).

ScrAutoFwd property (boolean)

Set to True (non-zero) to enable the “Screen Auto Forward” feature; otherwise set to False (zero).

ScreenPrintMode property (boolean)

Set to False (zero) for text mode, True (non-zero) for graphics mode. When this property is True, the PrinterColorMode and PrintScreenBackground properties affect how the screen is printed.

ScrMode property (integer)

Set to zero for normal (80 column) mode, 1 for extended (132 column) mode.

[†] Do not use in scripts

ScrollMode property (integer)

This controls the appearance of the vertical scroll bar: 0=no scroll bar (scroll bar is only visible if the current number of Rows will not fit in the current window size), 1=always show scroll bar, and 2=automatically show scroll bar when cursor is positioned near right border of window.

SlavePrintMode property (boolean)

Set to False (zero) for text mode, True (non-zero) for graphics mode. When this property is True, the PrinterFont..., PrinterOrientation, PrinterPaperSize and PrinterPaperSource properties affect how the slave (Aux) print job is printed.

Sound property (string)

This property may be used to specify a custom sound for the terminal beep. Set this property to null to use the default sound. Set to a valid .wav file name to play the specified wave file. Set to "SystemDefault", "SystemHand", "SystemExclamation" or "SystemAsterisk" to use the sound associated with a system event (as defined in the Control Panel Sounds applet). Set to *frequency,duration* to use a true "beep" at the specified frequency (Hz) and duration (ms).

StopBits property (integer)

This is the number of stop bits used by the serial port attached to the session. This property is only meaningful when the communication device type is "Serial Port". Acceptable values are 1 or 2. After changing this property, use the Reset method for the change to take effect.

Strip8th property (boolean)

Set to True (non-zero) to cause AccuTerm to truncate received data to 7 bits; otherwise set to False (zero).

TelnetAltBreak property (boolean)

Set to True (non-zero) to send the Telnet Break command when the **Break** key is pressed; otherwise set to False (zero) for to send the Telnet Interrupt Process command.

TelnetBinary property (boolean)

Set to True (non-zero) to enable Telnet binary communication mode; otherwise set to False (zero) for text communication mode.

TelnetBypass property (boolean)

Set to True (non-zero) to bypass initial option negotiation; otherwise set to False (zero) for normal negotiation. This is required for certain hosts like Pick's D3 which do not implement the complete Telnet protocol.

TelnetEcho property (boolean)

Set to True (non-zero) to enable Telnet remote-echo mode; otherwise set to False (zero) for local echo mode. Note: when using local echo, you should also set the Duplex property to "half".

TermType property (integer)

This is the terminal emulation setting for the session. Possible values are `atTermTTY` (basic TTY), `atTermVPA2` (ADDS Viewpoint A2), `atTermVP60` (ADDS Viewpoint 60), `atTermP60` (ProComm Viewpoint 60), `atTermA2E` (ADDS Viewpoint A2 Enhanced), `atTermWY50` (Wyse 50), `atTermWY60` (Wyse 60), `atTermVT52` (DEC VT-52), `atTermVT220` (DEC VT-220), `atTermANSI` (ANSI BBS), `atTermPICKMON` (Pick PC Monitor) or `atTermTEK` (Tektronix 4014).

XmodemTimeout property (integer)[†]

Specifies the number of seconds before Xmodem file transfer operations time out.

YmodemTimeout property (integer)[†]

Specifies the number of seconds before Ymodem file transfer operations time out.

ZmodemTimeout property (integer)[†]

Specifies the number of seconds before Zmodem file transfer operations time out.

ZmodemAuto property (integer)[†]

Set to True (non-zero) to enable automatic Zmodem downloads, otherwise set to False (zero).

[†] Do not use in scripts

The ScreenBlock Object

The `ScreenBlock` object is created by the `Session.GetBlock` method, and is used to preserve and restore the contents of a portion of the terminal screen. The `Session.SetBlock` method restores the contents of the `ScreenBlock` object back onto the terminal screen.

The Server Object

When a session is in “server” mode (see `ServerMode` property), a client application can obtain a `Server` object from the session. Using this `Server` object, the client can pass requests to the host server program and retrieve the results of the request. A session can create an unlimited number of `Server` objects, and requests from multiple `Server` objects are queued and processed in the order received.

`Description` property (string)

This read-only property returns the name of the host server. This is the same as the `Session.ServerName` property.

`ErrMsg` property (string)

This read-only property is the error message text returned from the host server, or from the local `Server` object in the case of invalid use of `Server` methods.

`ErrNum` property (long)

This read-only property is the error message number returned from the host server, or from the local `Server` object in the case of invalid use of `Server` methods.

`Notify` event

`object.Notify(code as Long)`

This event is fired when the request has been completed (or an error has occurred). If this event is not used to signal that the result is ready, then a “DoEvents” loop should be entered, testing the value of the `State` property, until the `State` is 1 (idle).

Request method

rc = **server**.Request (**command**)

This method returns True (non-zero) if the **command** has been queued for processing by the host server. Otherwise it return False, and the ErrMsg and ErrNum properties should be examined.

Result property (string)

This read-only property returns the result from the last request.

State property (integer)

This read-only property returns the server state: 0 = not in server mode, 1 = idle (result is ready, or ready for request), 2 = request queued, 3 = request being processed.

The Menu Object

Both the AccuTerm object and each session object contain a Menu object. The menu object is the root object of AccuTerm's menu structure. When no sessions are open, the menu object member of the AccuTerm object is the operational menu. When any session is open, the menu object member of the active session is the operational menu. The menu object contains a number of MnuBand objects, which contain either MnuTool objects or sub-menus (other MnuBand objects).

AddNew method (MnuBand object)

Set **MnuBandObject** = **MenuObject**.AddNew (**name**)

This method creates a new MnuBand object in the specified Menu object. After creating a new MnuBand object, use the MnuBand.AddNew method to create MnuTool objects for the newly created band.

Count property (integer)

This read-only property is the number of MnuBand objects contained in the menu.

MnuBands property (Collection of MnuBand objects)

This read-only property returns the collection of MnuBand objects contained in the menu.

RecalcLayout method

This method recalculates the menu band. Use this method after modifying properties of any contained bands or tools.

Remove method

MenuObject.Remove *name*

This method removes the specified band from the menu.

The MnuBand Object

A Menu object contains zero or more MnuBand objects. Examples of MnuBand objects are the main menu, the toolbar, the status bar, popup menus and sub-menus of the main menu or a popup menu. A MnuBand object contains either MnuTool objects or sub-menus (other MnuBand objects).

AddNew method (MnuTool object)

Set **MnuToolObject** = **MnuBandObject**.AddNew(*ID*)

This method creates a new MnuTool object with tool ID *ID* in the specified MnuBand object.

BandType property (integer)

This property is the type of band: 0 = toolbar or status bar, 1 = main menu, 2 = popup menu or sub-menu.

Caption property (string)

This property is used for sub-menus, and is the caption displayed in the higher-level menu item.

Count property (integer)

This read-only property is the number of MnuTool objects contained in the band.

DockingArea property (integer)

This property specifies the position of the toolbar or status bar: 1 = top, 2 = bottom, 4 = left, 8 = right.

Insert method

MnuBandObject.Insert MnuToolObject, AfterID

This method inserts a `MnuTool` object at a specified location in the band. ***AfterID*** is the tool ID to insert the tool after, and is a string.

`MnuTools` property (Collection of `MnuTool` objects)

This read-only property returns the collection of `MnuTool` objects contained in the menu band.

Name property (string)

This read-only property is the name of the `MnuBand` object.

Remove method

MnuBandObject.Remove ID

This method removes a `MnuTool` object from the band. ***ID*** is the tool ID to remove, and is a string.

Visible property (string)

This property is `True` if the band is visible.

The `MnuTool` Object

The `MnuTool` object is at the lowest level of the `Menu` object hierarchy, and contains properties for the caption, icon, state and action of a menu item, toolbar button or status bar field.

Action property (string)

This property contains the action to perform when the menu item or button is clicked. All built-in menu items have a default action associated with them, but an optional action can be specified by setting this property. Set this property to null to perform the default action. Otherwise, to execute a macro, enclose the macro statement in square brackets ([]). To transmit a character sequence to the host, set this property to the desired sequence.

`BeginGroup` property (integer)

Set this property to `True` to place a separator between this tool and the one before it.

Caption property (string)

This property is the caption displayed in the higher-level menu item. The caption is not displayed in a toolbar.

Checked property (integer)

Set this property to True to display a checked box next to the caption in popup menus or sub-menus.

Enabled property (integer)

Set this property to True to enable a menu item. If this property is False, the item is disabled and is displayed in a gray color.

Height property (long)

To set the height of the menu item to a specific size, set this property to the desired height in pixels. If this property is zero, the default height is used.

ImageNo property (integer)

Set this property to the ID of the icon to display in the menu or toolbar of the menu tool.

Style property (integer)

This property is type of tool: 0 is normal (icon only if a toolbar, otherwise icon and caption), 1 is caption only, 2 is icon only, and 3 is both caption and icon.

SubBand property (string)

This property is the name of the MnuBand object to be opened as a sub-menu when the tool is clicked.

ToolID property (long)

This property is the tool ID.

ToolTipText property (string)

This property is the text which is displayed in a floating window when the cursor is paused over a menu item or toolbar button.

Visible property (string)

This property is True if the tool (menu item or toolbar button) is visible.

Width property (long)

To set the width of the menu item to a specific size, set this property to the desired width in pixels. If this property is zero, the default width is used.

ACCUTERM PROGRAMMING

AccuTerm 97 contains several “private” commands which may be sent by host application programs. These commands allow for remote control of file transfer and data capture, executing DOS commands, enabling the mouse, displaying images, and programming the function and keypad keys. These commands are valid from any terminal emulation, and are described below:

ESC STX < command CR

Executes the DOS command **command**, then returns to AccuTerm emulation mode immediately. Command is executed concurrently with the terminal session.

ESC STX > command CR

Executes the DOS command **command** and suspends the terminal session until the command completes.

ESC STX 0 Disable mouse input; turns off mouse cursor.

ESC STX 1 Enable mouse input; turns on mouse cursor. Transmits mouse location whenever a mouse button is pressed. The format of the mouse location depends on terminal emulation:

ASCII: **STX b CR ccc . rr CR**

ANSI 7 bit: **ESC [n ~ ESC [r ; c R**

ANSI 8 bit: **CSI n ~ CSI r ; c R**

ASCII: b indicates which mouse button was pressed (**p**=left, **q**=right, **r**= center single click, **P**=left, **Q**=right, **R**=center double click), **ccc** is the three digit column of the mouse cursor and **rr** is the two digit row of the mouse cursor (both in decimal, 000 . 00 is the upper left corner). If **b** is lower case, the mouse was clicked once, **b** is uppercase if double clicked.

ANSI: n indicates which mouse button was pressed (101=left, 102=right, 103=center single click, 111=left, 112=right, 113=center double click), **r** is the row of the mouse cursor and **c** is the column of the mouse cursor (both in decimal, where 1 ; 1 is the upper left corner).

ESC STX D *p o m* ; *path* CR

Download file from host to PC. Protocol ***p*** may be A (ASCII), K (Kermit), X (Xmodem), Y (Ymodem) or Z (Zmodem); Overwrite ***o*** may be O (overwrite) or N (no overwrite) or, if protocol is Z, overwrite may be R (); Mode ***m*** may be T (text) or B (binary). ***Path*** is the drive, directory and file name of the file being received. When using Kermit, Ymodem or Zmodem protocols, only drive and directory need be specified, as the file name is included in the transfer protocol; however if the file name is specified here, it overrides the file name included in the transfer protocol.

ESC STX U *p m* ; *path* CR

Upload file from PC to host. Protocol ***p*** may be A (ASCII), K (Kermit), X (Xmodem), Y (Ymodem) or Z (Zmodem); Mode ***m*** may be T (text) or B (binary). ***Path*** is the drive, directory and file name of the file to send to the host. Wildcard characters (* or ?) are valid when using Kermit, Ymodem or Zmodem protocols.

ESC STX S Returns status of last file transfer. Status message is:

Status: *s* files *f* bytes *b* CR

where ***f*** is the number of files transferred, ***b*** is the number of bytes transferred, and ***s*** is the transfer status:

- 0 = transfer successful
- 1 = unable to open file
- 2 = transfer aborted by operator
- 3 = file already exists
- 4 = terminated due to timeout
- 5 = terminated due to corrupted data
- 6 = invalid packet type
- 7 = terminated by remote program
- 8 = 8 data bits required for protocol
- 9 = software flow control not allowed for protocol

ESC STX C o p ; path CR

Begin capture. Mode **o** may be O (overwrite), A (append), N (new file only) or C (clipboard). Source **p** may be P to capture printed data or null to capture received data. **Path** is the drive, directory and file name where the captured data is to be stored. All characters received (or printed) are stored in the file (or clipboard) until capturing is disabled (via local or remote command). Note: when capturing to the clipboard, **path** is ignored).

ESC STX C X

End capture. The file containing the captured data is closed.

ESC STX I Returns AccuTerm release, serial number, type and licensee information. Message format is:

ACCUTERM/WIN **rel serial type licensee... CR**

where **rel** is the AccuTerm release number, **serial** is the program serial number, **type** is SINGLE, SITE, CORP, DEALER or DEMO, and **licensee** is the name under which the program has been licensed.

ESC STX X Terminates AccuTerm session. If only one session exists, then AccuTerm is terminated also.

ESC STX W Saves the current session settings to disk. If no session file name exists, then one will be prompted for.

ESC STX E Selects extended (132 column) video mode as defined by the extended columns and rows in the **Screen Settings** category in the **Settings** dialog box.

ESC STX N Selects normal (80 column) video mode as defined by the normal columns and rows in the **Screen Settings** category in the **Settings** dialog box.

ESC STX Y path CR

Send clipboard or file to host. If **path** is null, the contents of the clipboard are sent, otherwise the specified file is sent. Each line is terminated as specified in the ASCII settings (**File Transfer Settings**). A **SUB** (CTRL-Z) is sent after the last line.

ESC STX P *command* CR

Executes macro command ***command***. See section on “**Scripting**” for details on the available commands. Multiple statements may be executed; separate each statement with **LF** or **EM** control characters. Note: you can call subroutines or functions contained within the script loaded in the main script window. For example, to call subroutine FOO, simply send: **ESC STX P FOO CR**. (Note: spaces are not part of the sequence, they are only shown here for clarity).

ESC STX F *t s k data* CR

Program function and keypad keys. Type ***t*** may be N (normal function keys), C (control function keys), A (ALT function keys) or K (keypad keys). Shift ***s*** may be U (unshifted) or S (shifted). Key code ***k*** may be digits 0 to 9 or : or ; according to the following table. ***Data*** contains the function key sequence data. Control codes may be represented by prefixing the a letter or symbol with ^ (e.g. to program a carriage return, enter ^M). To program a ^, enter ^^. To program a **CTRL+^**, enter ^~. Note: if ***data*** is enclosed in brackets ([]), then when the key is pressed, it will be interpreted as a macro command, rather than sent to the host.

Key Code	Function key	Editing key
0	F1	BKSP
1	F2	TAB
2	F3	INS
3	F4	DEL
4	F5	HOME
5	F6	END
6	F7	PGUP
7	F8	PGDN
8	F9	LEFT
9	F10	RIGHT
:	F11	UP
;	F12	DOWN
<		ESC
=		ENTER
>		KPD ENTER

For example, to program the **END** key to send the word "END", followed by a carriage return, the following sequence would be used:

ESC STX F K U 5 E N D ^ M CR

ESC STX iL , filename , col , row , width , height , aspect, border CR

Displays the image file **filename** at column **col** and row **row**. **Height** and **width** are optional; if specified (and not zero), the image is scaled to **height** rows and **width** columns. Otherwise, the original image size is used. If **aspect** is non-zero, the image aspect ratio is preserved (the specified width or height is reduced to preserve the aspect ration). **Border** is N for no border, B for simple border, R for raised border or I for inset border style. Image file types supported include bitmap, Windows metafile, JPEG, TIFF and TARGA. Images are considered "protected" data; a "clear un-protected characters" command will not clear images from the terminal screen.

ESC STX iD , filename CR

Removes the displayed image file **filename** from the screen.

ESC STX iC CR

Clears all displayed images from the screen.

ESC STX m filename CR

Plays MIDI sound file **filename**.

ESC STX w filename CR

Plays Wave sound file **filename**.

ESC STX h CR

Resets the mouse pattern table.

ESC STX h button HT pattern HT click HT dblclk CR

Adds an entry to the mouse pattern table.

ESC STX A *color* CR

Sets the foreground (text) color to ***color***.

ESC STX B *color* CR

Sets the background color to ***color***.

<u>Color</u>	<u>Display</u>
0	Black
1	Blue
2	Green
3	Cyan
4	Red
5	Magenta
6	Brown
7	Light Gray
8	Dark Grey
9	Light Blue
10	Light Green
11	Light Cyan
12	Light Red
13	Light Magenta
14	Yellow
15	White

Note: the actual display color may vary from the color shown in the table, since it is possible to modify the palette, either using the Color Settings section of the Settings dialog, or from code.

WYSE PROGRAMMING

This section describes the command sequences for programming the Wyse 50, Wyse 60 and ADDS Viewpoint Enhanced terminal emulations. These three emulations use a common command set with a few differences. The differences are noted.

Operating Modes

ESC ` *n* Set mode *n*. This command is used to set many of the terminal's operating modes. The mode *n* values and their function are shown in the table below:

<u><i>n</i></u>	<u>Function</u>
0	Cursor off
1	Cursor on
2	Block cursor
3	Line cursor
A	Normal protect character
6	Reverse protect character
7	Dim protect character
8	Screen off
9	Screen on
:	80 column screen
;	132 column screen
B	Protect blink on
C	Protect invisible on
E	Protect underline on
F	Protect reverse on
G	Protect dim on

ESC B Places terminal in local mode.

ESC C or ESC D F
Places terminal in full duplex mode.

ESC D H	Places terminal in half duplex mode.
ESC N	Disables auto scrolling. Normally if the cursor is moved down below the last line, the screen is scrolled up one line. If auto scrolling is disabled, the cursor moves to the top line and no scrolling takes place.
ESC O	Enables auto scrolling.
ESC X or ESC u	Turns the monitor mode off.
ESC U	Turns the monitor mode on.
ESC k	Turns the local edit submode on. This command is not supported by AccuTerm and is ignored. [†]
ESC l	Turns the local edit submode off. This command is not supported by AccuTerm and is ignored.
ESC q	Turns the insert mode on. In this mode, all characters sent to the screen are inserted into the line with any existing characters moved one column to the right.
ESC r	Turns the insert mode off.
ESC # or SI	Lock keyboard. [‡]
ESC 5 or EOT	Lock keyboard. [‡]
ESC " or SO	Unlock keyboard. [†]
ESC 6 or STX	Unlock keyboard. [‡]
GS	If the graphics mode is enabled, this command puts the terminal in the Tektronix emulation mode. If the graphics mode is disabled, this command is ignored.

[†] Only valid in Wyse 50 and Wyse 60 modes.

[‡] Only valid in Viewpoint Enhanced mode.

ESC 1 ESC FF

If the graphics mode is enabled, this command puts the terminal in the Tektronix emulation mode. If graphics mode is disabled, it sets a tab stop at the current cursor column.

ESC % ! 0 Enters Tektronix 4100 graphics mode.

CAN Exits from the Tektronix graphics emulation mode.

CAN Turns on the cursor.[‡]

ETB Turns off the cursor.[‡]

ESC 2 Exits from the Tektronix graphics emulation mode. If the terminal is not in the graphics emulation mode, it clears any tab stops at the current cursor column.

ESC F message CR

Displays a message in the host status line. The message string can be up to 46 characters in 80 column mode and 98 characters in the 132 column mode.

ESC C ; message EM

Programs the answerback message (up to 30 characters).[§]

ESC C < Sends the answerback message to the host, followed by **ACK**. If no answerback message has been programmed, simply sends **ACK**.[§]

ESC C = Erases the answerback message.[§]

ESC C D Selects the primary character set.[§]

ESC C E Selects the secondary character set.[§]

ESC d / End of line wrap on. When the cursor reaches the end of a line, it will wrap to the beginning of the next line.[§]

ESC d . End of line wrap off. When the cursor reaches the end of a line, it does not wrap to the beginning of the next line.[§]

[‡] Only valid in Viewpoint Enhanced mode.

[§] Only valid in Wyse 60 mode.

ESC ~ SPACE

Enhance mode off. If the current emulation is ADDS VPA2E, then the emulation will change to ADDS VPA2.

ESC ~ ! Enhance mode on. If the current emulation is ADDS VPA2, then the emulation will change to ADDS VPA2E.

ESC ~ " Select Wyse 50 emulation.

ESC ~ % Select ADDS VPA2E emulation.

ESC ~ 4 Select Wyse 60 emulation.

ESC ~ 6 Select VT-52 emulation.

ESC ~ ; or **ESC ~ <** or **ESC ~ =** or **ESC ~ A** or **ESC ~ B**
Select VT-220 emulation.

ESC ~ > Select Tektronix 4014 emulation.

Cursor Positioning

FF	Cursor right. If the cursor is on the last column of the line, the cursor will wrap to the next line, possibly scrolling the screen up. [†]
ACK	Cursor right. If the cursor is on the last column of the line, the cursor will wrap to the next line, possibly scrolling the screen up. [‡]
BS	Cursor left. If the cursor is at the beginning of the line, it is moved up to the last column of the previous line. If the cursor is at the home position, it is moved to the lower right hand corner of the screen.
NAK	Cursor left. Same as the BS command. [‡]
HT or ESC i	Moves the cursor to the next programmed tab stop. For this command to work, tab stops must be programmed with the ESC 1 command.
ESC I	Move the cursor left to the previous tab stop.
LF	Cursor down. If the cursor is on the last line of the screen and the "no scroll" mode is turned off, the screen will scroll up one line. Otherwise, the cursor will move to the top line of the screen.
VT	Cursor up. If the cursor is at the top row, it is moved to the bottom row. [†]
SUB	Cursor up. If the cursor is at the top row, it is moved to the bottom row. [‡]
ESC j	Cursor up. If the cursor is at the top row, the screen is scrolled down.

[†] Only valid in Wyse 50 and Wyse 60 modes.

[‡] Only valid in Viewpoint Enhanced mode.

CR	Moves the cursor to the first column (column zero) of the current row.
US	Moves the cursor down one row and over to the first column (column zero).
DEL	Ignored.
RS or ESC {	Moves the cursor to the home position (upper left corner of the screen).
VT r	Moves cursor to row r , where r is a valid row code from the Viewpoint Cursor Address Table (appendix B). [‡]
DLE c	Moves the cursor to column c where c is a valid column in the from the Viewpoint Cursor Address Table (appendix B). [‡]
ESC = r c	Moves the cursor to row r and column c . r and c are single byte cursor address codes from the Wyse Cursor Address Table (appendix A). Note that this command cannot position the cursor past column 95.
ESC Y r c	Moves the cursor to row r and column c where r and c are single byte cursor address codes from the Wyse Cursor Address Table (appendix A). Note that this command cannot position the cursor past column 95. [‡]
ESC a rr R ccc C	Moves the cursor to row rr and column ccc . rr is the two digit decimal number of the row (from row 1 at the top). ccc is the three digit decimal number of the column (from column 1 at the left). Note that this command can address the entire 132 column screen.
ESC X c	Moves the cursor to column c where c is a single byte column address from the Wyse Cursor Address Table (appendix A). Note that this command cannot position the cursor past column 95.

[‡] Only valid in Viewpoint Enhanced mode.

- ESC [*r*** Moves the cursor to row *r* where *r* is a single byte row address from the Wyse Cursor Address Table (appendix A).
- ESC K** Page forward. If currently on the last page, the first page is selected.[†]
- ESC J** Page back. If currently on the first page, the last available page is selected.
- ESC]** Selects page 0.
- ESC }** Selects page 1.
- ESC - *n r c*** Selects page *n* (pages numbered from 0 to 6) and positions the cursor to row *r*, column *c*. *r* and *c* are single byte cursor address codes from the Wyse Cursor Address Table (appendix A).
- ESC w *n*** Selects page *n*. Pages are numbered from 0 to 6.[§]
- ESC w C** Page forward. If currently on the last page, the first page is selected.[§]
- ESC w P** Page back. If currently on the first page, the last available page is selected.[§]
- ESC w @ *n r c*** Selects page *n* (pages numbered from 0 to 6) and positions the cursor to row *r*, column *c*. *r* and *c* are single byte cursor address codes from the Wyse Cursor Address Table (appendix A).[§]
- ESC x A *r*** Splits screen horizontally at row *r*, where *r* is a valid row code from the Wyse Cursor Address Table (appendix A) or from the Viewpoint Cursor Address Table (appendix B) if in Viewpoint Enhanced mode.
- ESC x 1 *r*** Splits screen horizontally at row *r* and clear screen, where *r* is a valid row code from the Wyse Cursor Address Table (appendix A) or from the Viewpoint Cursor Address Table (appendix B) if in Viewpoint Enhanced mode..

[†] Only valid in Wyse 50 and Wyse 60 modes.

[§] Only valid in Wyse 60 mode.

- ESC x @** Redefine screen as one window.[§]
- ESC x 0** Redefine screen as one window and clear screen.[§]

Erasing and Editing

- SOH** Clears the screen and returns the cursor to the home position (upper left corner of the screen).[‡]
- FF** Clears the screen and returns the cursor to the home position (upper left corner of the screen). Turns off the protected mode.[‡]
- SUB** Clears the screen and returns the cursor to the home position (upper left corner of the screen).[†]
- ESC * or ESC +**
 Clears the screen and move the cursor to the home position (upper left corner of the screen). Turns off the protect mode.
- ESC T or ESC t**
 Clears from the cursor position to the end of the current line.
- ESC K** Clears from the cursor position to the end of the current line.[‡]
- ESC Y or ESC y**
 Clears from the cursor position to the end of the screen.[†]
- ESC k** Clears from the cursor position to the end of the screen.[‡]
- ESC ,** Clears the screen to protected spaces. Turns off the protect mode. "No scroll" mode is turned off (auto scroll).

[‡] Only valid in Viewpoint Enhanced mode.

[†] Only valid in Wyse 50 and Wyse 60 modes.

ESC E Insert line. A blank line is inserted before the current line. The current line and all lines below are moved down one row. The bottom line of the screen is lost. The cursor is moved to the left most column of the inserted line.

ESC Q Insert character. A blank is inserted at the current cursor position. All characters from the cursor position to the right are moved right one column.

ESC R Delete line. The current line is deleted. All lines below the current line are moved up one position. The bottom line is blank. The cursor is moved to the left most column.

ESC W Delete character. The character at the current cursor position is deleted. All characters to the right of the cursor are moved left one column and a blank is placed on the last column of the line.

ESC C N w h
Box rectangle. Current cursor location is upper left, **w** and **h** specify rectangle width and height.

ESC C G r c
Box rectangle. Current cursor location is one corner, **r** and **c** specify row and column of other corner.

ESC C F r c x
Clear unprotected rectangle. Cursor location is one corner, **r** and **c** specify row and column of other corner. Rectangle is cleared to character **x**.

ESC C H r c x
Clear entire rectangle. Cursor location is one corner, **r** and **c** specify row and column of other corner. Rectangle is cleared to character **x**.

ESC C ^ a sp P rr1 R ccc1 C rr2 R ccc2 C dp P rr3 R ccc3 C
Copy / swap / move rectangle. Action **a** is: 0 to swap, 1 to copy, 2 to move. Source page **sp**, upper left corner **rr1**, **ccc1**; lower right corner **rr2**, **ccc2**; destination page **dp**, upper left corner **rr3**, **ccc3**. All parameters are decimal numbers.

Video Attributes

Video attributes work differently in the Wyse 50 or Viewpoint Enhanced emulations than they do in the Wyse 60 emulation. Under Wyse 60, the attributes are non-embedded. In other words, they do not take up a character position on the screen. Under Wyse 50 or Viewpoint Enhanced, the attribute takes up a space and changes all following characters to the end of the screen or until another attribute character is encountered.

Under Viewpoint Enhanced mode, a single non-embedded (takes up no space) "tagged" attribute may be assigned and used. This "tagged" attribute is also the "protect" attribute.

ESC 0 a Assigns the "tagged" video attribute. **a** is the video attribute code from the Viewpoint Attribute Code Table (appendix B).[‡]

SO Start tagged attribute. All characters received after this code are displayed with the currently assigned tagged attribute. If the tagged attribute is changed, the attribute of the displayed characters also changes.[‡]

SI End tagged attribute. All characters received after this code are displayed with the normal video attribute.[‡]

ESC A n a Sets the video attribute for any of the four application display areas. **n** is the display field code and **a** is the attribute code from the Wyse Attribute Code Table (Appendix A). The application display field codes are:

<u>n</u>	<u>Display area</u>
0	The main screen
1	The function key labeling line
2	The status line
3	The host message field

ESC G a Assign visual attribute. This command is used to start a specific attribute on the main screen. In Wyse 50 and ADDS VP mode, this command changes all characters from the

[‡] Only valid in Viewpoint Enhanced mode.

current position until the end of the screen or until another attribute code character is encountered. In Wyse 60 mode, this command selects the attribute for characters displayed after this command. The visual attribute is specified by the attribute code **a** from the Wyse Attribute Code Table (appendix A).

Protected Attributes

Protected attributes are used by some software to protect characters from being overwritten. If the terminal is running in its protected mode, the cursor cannot be positioned over protected characters and all unprotected characters can be cleared by one command.

- | | |
|--|--|
| ESC & | Enable protect mode and set "no scroll" mode. |
| ESC ' | Disable protect mode and set "auto scroll". |
| ESC) | Start protect mode. All characters sent after this sequence become protected characters until the protect mode is turned off. |
| SO | Start protect mode. [‡] |
| ESC (| Stop protect mode. All characters sent after this sequence are unprotected characters. |
| SI | Stop protect mode. [‡] |
| ESC ! a | Writes all unprotected attributes with a specified attribute where a is a valid attribute code from the Wyse Attribute Code Table (appendix A). |
| ESC . c | Replaces all unprotected characters with the character <i>c</i> . |
| ESC ; or ESC : or SUB | Clears all unprotected characters. |

[‡] Only valid in Viewpoint Enhanced mode.

- ESC ,** Clears the screen to protected spaces. The protect mode is turned off and the auto scroll function is turned on.
- ESC V** Clears the entire cursor column to protected spaces.

Line Graphics

- ESC H g** Display a line graphic character at the current cursor position. The graphic character is specified by graphic character code **g** from the Wyse Graphic Character Table (appendix A). The line graphics characters can be used for drawing simple boxes on the screen in text mode. It should not be confused with the Tektronix graphic mode which is much more sophisticated and capable of drawing pie charts, scientific diagrams, etc.
- ESC H STX** Turn on line graphics mode. All characters received while the line graphics mode is on are interpreted as graphics characters according to the Wyse Graphic Character Table (appendix A).
- ESC H ETX** Turn off line graphics mode.

Printer Control and Terminal Reports

- ESC SPACE** Reports the terminal identification to the host computer.
Sends 50 followed by a carriage return.
- ESC ?** Transmits the cursor address to the host computer. The cursor address is transmitted in 80 column format, followed by a **CR**.
- ESC / or ESC w** Transmits the page number and cursor address to the host computer. The cursor address is transmitted in 80 column format, followed by a **CR**.
- ESC 6** Sends the current row to the host computer followed by a carriage return.[†]
- ESC 7** Sends the entire screen to the host computer. Each row of the screen is followed by a carriage return.
- ESC L or ESC P or ESC p** Prints the entire screen to the printer port.
- ESC M** Sends the character at the current cursor position to the host computer.
- ESC S** Sends a message unprotected. This function is not supported by AccuTerm.
- ESC b** Sends the current cursor position in the 132 column format.
- ESC s** Sends a message. This function is not supported by AccuTerm.
- DC2** Auto print mode. Characters are displayed and printed. This command will not function if the printer selection is set to "No printer".

[†] Not valid in ADDS Viewpoint A2 mode.

- DC4** Cancel auto or transparent print mode. Note: This command will not turn the printer off if it was turned on by the Viewpoint Enhanced transparent print or ANSI print commands.
- CAN** Transparent print mode. Characters are printed, but not displayed.
- ESC 3** Transparent print mode. Characters are printed, but not displayed.[‡]
- ESC 4** Cancel transparent print mode.[‡]
- ESC d #** Transparent print mode. Characters are printed, but not displayed.[§]
- ESC [? 5 i**
Transparent print mode (ANSI). Characters are printed, but not displayed. When AccuTerm receives this command, it goes into transparent print mode until it receives the ANSI print off sequence below. This command is useful for printing graphics data to the printer. Since AccuTerm requires the specific ANSI cancel command below to exit transparent print mode, there is less chance of a control character interrupting the graphic printing.
- ESC [? 4 i**
Cancel transparent print mode (ANSI).

[‡] Only valid in Viewpoint Enhanced mode.

[§] Only valid in Wyse 60 mode.

Programming Function Keys

The Wyse 50, Wyse 60 and Viewpoint Enhanced emulations support the ability to download function key values from the host computer. The function key programming consists of two steps, downloading the actual values which the key will send to the host whenever it is pressed or downloading a descriptive function key label that is displayed on the function key labeling line.

When programming the function keys or function key labels, all characters (including control characters) may be included in the sequence, except for the terminator (**DEL** for keys, **CR** for labels).

To clear a programmed function key or label, send the same command used for programming the key or label, but omit the sequence.

ESC Z k sequence DEL

Program function key **k** to send **sequence** to host when pressed. The function key codes are specified in the Wyse Function Key Table (appendix A).

ESC Z f label CR

Program function key label field **f** as **label**. The field codes are specified in the Wyse Function Key Table (appendix A).

ESC Z 0 k sequence DEL

Program function or keypad key **k** to send **sequence** to host when pressed. The function and keypad key codes are specified in the Wyse Function Key Table (appendix A).

ESC Z (text CR

Sets the unshifted label line to **text**. If **text** is omitted, clears the unshifted label line.

ESC Z) text CR

Sets the shifted label line to **text**. If **text** is omitted, clears the shifted label line.

ESC C U Clear all redefinable key definitions to their default values.[§]

ESC Z ~ k Read programmable key definition for key **k**.

[§] Only valid in Wyse 60 mode.

ADDS PROGRAMMING

This section describes the command sequences for programming the ADDS Viewpoint A2, Viewpoint 60 and Procomm VP60 terminal emulations. These emulations use a common command set with a few differences. The differences are noted.

Operating Modes

ESC B	Status line off. [†]
ESC b	Status line on. [†]
ESC D	Screen display off. [†]
ESC d	Screen display on. [†]
ETB	Turns off the cursor.
CAN	Turns on the cursor.
GS	If the graphics mode is enabled, this command puts the terminal in the Tektronix emulation mode. If the graphics mode is disabled, this command is ignored.
ESC % ! 0	Enters Tektronix 4100 graphics mode.
CAN	Exits from the Tektronix graphics emulation mode.

[†] Not valid in ADDS Viewpoint A2 mode.

Cursor Positioning

- ACK** Cursor right. If the cursor is on the last column of the line, the cursor will wrap to the next line, possibly scrolling the screen up.
- BS** or **NAK** Cursor left. If the cursor is at the beginning of the line, it is moved up to the last column of the previous line. If the cursor is at the home position, it is moved to the lower right hand corner of the screen.
- LF** Cursor down. If the cursor is on the last line of the screen and the "no scroll" mode is turned off, the screen will scroll up one line. Otherwise, the cursor will move to the top line of the screen.
- SUB** Cursor up. If the cursor is at the top row, it is moved to the bottom row.
- CR** Moves the cursor to the first column (column zero) of the current row.
- VT *r*** Moves cursor to row *r*, where *r* is a valid row code from the Viewpoint Cursor Address Table (appendix B).
- DLE *c*** Moves the cursor to column *c* where *c* is a valid column in the from the Viewpoint Cursor Address Table (appendix B).
- ESC Y *r c*** Moves the cursor to row *r* and column *c* where *r* and *c* are single byte cursor address codes from the Wyse Cursor Address Table (appendix A). Note that this command cannot position the cursor past column 95.

Erasing and Editing

FF	Clears the screen and returns the cursor to the home position (upper left corner of the screen). Turns off the protected mode.
ESC K	Clears from the cursor position to the end of the current line.
ESC k	Clears from the cursor position to the end of the screen.
ESC M	Insert line. A blank line is inserted before the current line. The current line and all lines below are moved down one row. The bottom line of the screen is lost. The cursor is moved to the left most column of the inserted line. [†]
ESC F	Insert character. A blank is inserted at the current cursor position. All characters from the cursor position to the right are moved right one column. [†]
ESC 1	Delete line. The current line is deleted. All lines below the current line are moved up one position. The bottom line is blank. The cursor is moved to the left most column. [†]
ESC E	Delete character. The character at the current cursor position is deleted. All characters to the right of the cursor are moved left one column and a blank is placed on the last column of the line. [†]

[†] Not valid in ADDS Viewpoint A2 mode.

Video Attributes

The Viewpoint A2, Viewpoint 60 and Procomm VP60 have different ways of programming video attributes.

The ADDS Viewpoint A2 terminal supports video attributes through the use of "tagged" characters. When a character is received from the host, it is either tagged or normal. Tagged characters are displayed with the currently assigned tagged video attribute.

Programming video attributes involves two separate steps. First, the tag attribute must be assigned. Next, in order to display characters in the assigned tag attribute, a "start tag attribute" code must be sent. To send characters in the normal attribute, an "end tag attribute" code must be sent. These two steps are independent and can be executed in either order.

Any time the tagged attribute is changed, all of the tagged characters already on the screen will be displayed in the new attribute. However, only one video attribute can be displayed at a time. This limitation only exists in the standard Viewpoint emulation mode. In the Viewpoint enhanced mode, both the Viewpoint and Wyse attributes can be used at the same time.

The ADDS Viewpoint 60 (and Procomm VP60) emulations use the same escape sequence as the Viewpoint A2 but it is interpreted differently. Under ADDS Viewpoint 60 mode, the attribute takes up a space and changes all following characters to the end of the screen or until another attribute character is encountered. It works similar to the Wyse 50 video attributes. Under Procomm VP60, multiple visual attributes may be displayed at the same time, but the attribute character does not use a space on the screen, and does not affect any previously displayed characters. It works similar to the Wyse 60 video attributes.

SO Start tagged attribute. All characters received after this code are displayed with the currently assigned tagged attribute. If the tagged attribute is changed, the attribute of the displayed characters also changes.

SI End tagged attribute. All characters received after this code are displayed with the normal video attribute.

ESC 0 a **ADDS Viewpoint A2:** Assign visual attribute. This command sets the current visual attribute. All characters which are “tagged” are displayed on this attribute. The attribute is specified by the attribute code **a** from the Viewpoint Attribute Code Table (appendix B).

ADDS Viewpoint 60: Assign visual attribute. This command is used to start a specific attribute. This command changes all characters from the current position until the end of the screen or until another attribute code character is encountered. The attribute uses a screen position. The attribute is specified by the attribute code **a** from the Viewpoint Attribute Code Table (appendix B).

Procomm VP60: Assign visual attribute. This command is used to start a specific attribute. The attribute does not use a screen position. All characters output after this command are displayed in the specified attribute. The attribute code is specified by the attribute code **a** from the Viewpoint Attribute Code Table (appendix B).

Line Graphics

ESC 1 Turn on line graphics mode. All characters received while the line graphics mode is on are interpreted as graphics characters according to the Viewpoint 60 Graphic Character Table (appendix B).[†]

ESC ? Turn off line graphics mode.[†]

[†] Not valid in ADDS Viewpoint A2 mode.

Printer Control

- DC2** Auto print mode. Characters are displayed and printed. This command will not function if the printer option on the setup menu is set to "None".
- DC4** Cancel auto or transparent print mode. Note: This command will not turn the printer off if it was turned on by the Viewpoint Enhanced transparent print or ANSI print commands.
- ESC 3** Transparent print mode. Characters are printed, but not displayed.
- ESC 4** Cancel transparent print mode.
- ESC [? 5 I** Transparent print mode (ANSI). Characters are printed, but not displayed. When AccuTerm receives this command, it goes into transparent print mode until it receives the ANSI print off sequence below. This command is useful for printing graphics data to the printer. Since AccuTerm requires the specific ANSI cancel command below to exit transparent print mode, there is less chance of a control character interrupting the graphic printing.
- ESC [? 4 I** Cancel transparent print mode (ANSI).

ANSI PROGRAMMING

The AccuTerm ANSI emulations provide DEC VT-220, VT-52 and ANSI.SYS device driver (BBS) emulations. The VT-220 emulation supports international characters, 7 and 8 bit control codes, programmable function keys and a complete VT-220 command set. The ANSI emulation supports the PC character set and the attribute and cursor commands supported by the DOS ANSI.SYS device driver (AccuTerm does not use the ANSI.SYS driver). The ANSI emulation only supports 7 bit escape sequences. Commands which are recognized by the ANSI emulation are indicated.

AccuTerm in VT-220 mode will respond to 7-bit and 8-bit control codes (hex 00-1F and 80-9F). For convenience, most command sequences which may use 8-bit control codes are documented using the 8-bit control code. For every 8-bit control code, an equivalent 7-bit escape sequence may also be used, as shown in the following table:

CSI	=	ESC [
DCS	=	ESC P
ST	=	ESC \

The AccuTerm VT-220 emulation is capable of operating in a variety of modes. Some of the default operating modes are determined by settings in the AccuTerm configuration. Those modes are shown below:

VT220 / VT52: VT-220 emulation is selected by setting the setup configuration terminal to "DEC VT220". VT-52 emulation is selected by "DEC VT52".

Columns: 80 column mode is selected by setting the setup configuration screen text mode to "Normal"; 132 column mode is selected by "Extended".

The following operating modes are not determined by settings in the AccuTerm configuration, but may be changed by sending the appropriate ANSI command sequence:

8-bit controls: default is 7-bit.

Keyboard: default is unlocked.

Insert/Replace: default is replace.

Line feed / New line: default is line feed.

Cursor keys: default is cursor.

Origin: default is absolute.

Auto wrap: default is on.

Tabs: default is every 8 columns.

Cursor: default is on.

Print extent: default is full screen.

Printer form-feed: default is off.

Graphics mode: default is off.

AccuTerm VT-220 emulation supports 5 character sets: ASCII, International, ISO-Latin1, Graphics and Scientific. The default character set assignment is G0=ASCII, G1=ASCII, G2=International and G3=ASCII. The GL set (hex 20-7E) defaults to G0 (ASCII) and the GR set (hex A0-FE) defaults to G2 (International).

Operating Modes

ESC SPACE F or **CSI ; 1 " p**
Causes AccuTerm to send 7-bit control codes **ESC** [and **ESC** O instead of the 8-bit control codes **CSI** and **SS3**.

ESC SPACE G or **CSI ; 2 " p**
Causes AccuTerm to send 8-bit control codes **CSI** (hex 9B) and **SS3** (hex 8F).

CSI 2 h Locks the keyboard.[†]

CSI 2 l Unlocks the keyboard.[†]

CSI 4 h Insert mode on.[†]

CSI 4 l Insert mode off.[†]

CSI 12 h Full duplex (no local echo).[†]

CSI 12 l Half duplex (local echo).[†]

CSI 20 h Process **LF**, **VT** and **FF** as "new line"; that is perform carriage return and line feed.[†]

CSI 20 l Process **LF**, **VT** and **FF** as line feed.[†]

CSI ? 1 h Cursor keys return application codes.

CSI ? 1 l Cursor keys return cursor codes.

CSI ? 2 l Enter VT52 emulation mode.

CSI ? 3 h Extended video mode (132 columns).[†]

CSI ? 3 l Normal video mode (80 columns).[†]

[†] This command is valid in ANSI mode.

CSI ? 6 h Causes cursor positioning to be relative to the currently defined scrolling region.

CSI ? 6 l Causes cursor positioning to be absolute (not relative).

CSI ? 7 h Sets autowrap mode. When the cursor is on the last character of a line, receipt of another character causes the cursor to move to the first column of the next line.[†]

CSI ? 7 l Resets autowrap mode. The cursor will not move past the last column of the line upon receipt of another character.[†]

CSI ? 9 h or **CSI 0 \$ ~**
Status line off.

CSI ? 9 l Status line on.

CSI 1 \$ ~ Display local status line.

CSI 2 \$ ~ Display host-writable status line.

CSI 0 \$ } Data sent to screen's data area.

CSI 1 \$ } Data sent to host-writable status line.

CSI ? 18 h
Causes a form-feed character (hex 0C) to be sent to the printer after each print screen.[†]

CSI ? 18 l
No character is sent to the printer after each print screen.[†]

CSI ? 19 h
Print screen command causes the full screen to be printed.

CSI ? 19 l
Print screen command only prints the currently defined scrolling region.

CSI ? 25 h
Cursor on.[†]

[†] This command is valid in ANSI mode.

CSI ? 25 1
Cursor off.[†]

CSI ? 38 h or **GS** or **ESC 1**
Enters Tektronix 4014 graphics mode.[†]

CSI ? 38 1 or **CAN** or **ESC 2**
Exits Tektronix 4014 graphics mode.[†]

ESC % ! 0Enters Tektronix 4100 graphics mode.[†]

ESC = Numeric pad returns application codes.

ESC > Numeric pad returns numbers.

Character Set Selection

ESC (B Sets character set G0 to ASCII (default).

ESC) B Sets character set G1 to ASCII (default).

ESC * B Sets character set G2 to ASCII.

ESC + B Sets character set G3 to ASCII.

ESC (< Sets character set G0 to International.

ESC) < Sets character set G1 to International.

ESC * < Sets character set G2 to International (default).

ESC + < Sets character set G3 to International.

ESC (A Sets character set G0 to ISO-Latin1.

ESC) A Sets character set G1 to ISO-Latin1.

ESC * A Sets character set G2 to ISO-Latin1.

[†] This command is valid in ANSI mode.

ESC + A Sets character set G3 to ISO-Latin1.

ESC (0 Sets character set G0 to graphics.

ESC) 0 Sets character set G1 to graphics.

ESC * 0 Sets character set G2 to graphics.

ESC + 0 Sets character set G3 to graphics.

ESC (S Sets character set G0 to scientific.

ESC) S Sets character set G1 to scientific.

ESC * S Sets character set G2 to scientific.

ESC + S Sets character set G3 to scientific.

SI Invoke character set G0 into GL (the GL set corresponds to character codes in the range of hex 20-7E); this is the default.

SO Invoke character set G1 into GL.

ESC ~ Invokes character set G1 into GR (the GR set corresponds to character codes in the range of hex A0-FE).

ESC n Invokes character set G2 into GL.

ESC } Invokes character set G2 into GR (default).

ESC o Invokes character set G3 into GL.

ESC | Invokes character set G3 into GR.

SS2 or ESC N Invokes character set G2 into GL for the next character only.

SS3 or ESC O Invokes character set G3 into GL for the next character only.

- ESC C** Hard reset. Re-reads AccuTerm configuration file, then resets all operating modes and character sets to their default values. Clears the screen and I/O buffer.
- CSI ! p** Soft reset. Resets all operating modes and character sets to their default values.

Cursor Positioning

- BS** Moves the cursor back one space. If the cursor is at the beginning of the line, no action occurs.[†]
- HT** Moves the cursor to the next programmed tab stop. If there are no more tab stops, the cursor moves to the right margin.[†]
- LF** Moves the cursor down one line. If the cursor is on the last line of the scrolling region, the screen will scroll up one line.[†]
- VT** Same as **LF**.[†]
- FF** Same as **LF**.[†]
- CR** Moves the cursor to the left margin of the current line.[†]
- NEL or ESC E** Moves the cursor to the first column of the next line of the scrolling region. If the cursor is on the last line of the scrolling region, the screen will scroll up one line.
- IND or ESC D** Moves cursor down one line in the same column. If the cursor is on the last line of the scrolling region, the screen will scroll up one line.
- RI or ESC M** Moves the cursor up one line in the same column. If the cursor is on the first line of the scrolling region, the screen will scroll down one line.

[†] This command is valid in ANSI mode.

CSI n B or **CSI n e**

Moves cursor down **n** lines in the same column.[†]

CSI n E Moves cursor down **n** lines and to the first column.

CSI n A Moves the cursor up **n** lines in the same column.[†]

CSI n F Moves the cursor up **n** lines and to the first column.

CSI n D Moves cursor left **n** columns.

CSI n C or **CSI n a**

Moves cursor right **n** columns.[†]

CSI line ; column H or **CSI line ; column f**

Move cursor to line **line**, column **column**.[†]

CSI n d Moves cursor to line **n**.

CSI n G or **CSI n `**

Moves cursor to column **n**.

HTS or **ESC H** or **CSI 0 W**

Sets a tab stop at the column where the cursor is.

CSI 0 g or **CSI 2 W**

Clears a tab stop at the column where the cursor is.

CSI 3 g or **CSI 5 W**

Clears all tab stops.

CSI ? 5 W Sets tab stops every 8th column.

CSI n I Move forward **n** tab stops.

CSI n Z Move backward **n** tab stops.

[†] This command is valid in ANSI mode.

CSI n V or **CSI n SPACE R**

Display a preceding page. If **n** is 0 or 1, the previous page is displayed, otherwise **n** specifies the number of pages back to be displayed.

CSI n U or **CSI n SPACE Q**

Displays a following page. If **n** is 0 or 1, the next page is displayed, otherwise **n** specifies the number of pages forward to be displayed.

CSI n SPACE P

Display page **n**.

CSI top ; bottom r

Set scrolling region. The first line of the scrolling region is set to **top**; the last line to **bottom**. Default values for **top** and **bottom** are 1 and 24 respectively. Once the scrolling region is defined, if origin mode is set to relative, the cursor may be positioned into, but not out of, the scrolling region.

ESC 7 or **CSI s**

Save state (cursor position, video attribute, character set, autowrap, origin mode and protect mode).[†]

ESC 8 or **CSI u**

Restore state.[†]

Erasing and Editing

CSI 0 J Erases from the cursor to the end of the screen, including the cursor position.

CSI 1 J Erases from the beginning of the screen to the cursor, including the cursor position.

CSI 2 J Erases the entire screen (the cursor position is not moved).[†]

[†] This command is valid in ANSI mode.

CSI 0 K	Erases from the cursor to the end of the line, including the cursor position. [†]
CSI 1 K	Erases from the beginning of the line to the cursor, including the cursor position.
CSI 2 K	Erases the entire line (the cursor is not moved).
CSI ? 0 J	Erases all unprotected characters from the cursor to the end of the screen, including the cursor position.
CSI ? 1 J	Erases all unprotected characters from the beginning of the screen to the cursor, including the cursor position.
CSI ? 2 J	Erases all unprotected characters on the entire screen (the cursor position is not moved).
CSI ? 0 K	Erases all unprotected characters from the cursor to the end of the line, including the cursor position.
CSI ? 1 K	Erases all unprotected characters from the beginning of the line to the cursor, including the cursor position.
CSI ? 2 K	Erases all unprotected characters on the line (the cursor is not moved).
CSI n X	Erases n characters, beginning with the current cursor position.
CSI n @	Insert n blank characters beginning at the current cursor position.
CSI n L	Insert n blank lines beginning at the cursor line.
CSI n P	Delete n characters beginning at the current cursor position.
CSI n M	Delete n lines beginning at the cursor line.

[†] This command is valid in ANSI mode.

Video Attributes

CSI n ; n . . . m

Selects video attributes and/or character foreground and background colors according to the ANSI Attribute Code Table (Appendix C). Characters received after this command are displayed in the selected video attribute. If multiple parameters are used, their effects are cumulative (e.g. 0;4;5 selects blinking-underlined). A parameter value of 0 resets all attributes.[†]

CSI 0 " q Unprotected mode. Characters received after this command are erasable using the "erase unprotected characters" command.

CSI 1 " q Protected mode. Characters received after this command are not erasable using the "erase unprotected characters" command.

Printer Control and Terminal Reports

CSI 0 i Prints the screen display. Either the full screen or scrolling region may be selected, and a form feed may be sent after printing (see Operating Modes section).

CSI ? 1 i Prints the current cursor line.

CSI 5 i Transparent print mode. Characters are printed, but not displayed.

CSI ? 5 i Auto print mode. Characters are displayed and printed.

CSI 4 i or **CSI ? 4 i**
Cancel transparent or auto print mode.

[†] This command is valid in ANSI mode.

CSI 0 c or **ESC Z**

Request primary device attributes. AccuTerm will respond:

CSI ? 6 2 ; 1 ; 2 ; 6 ; 8 c

CSI > 0 c Request secondary device attributes. AccuTerm will respond:

CSI > 1 ; 2 0 ; 0 c

CSI 5 n Request terminal status. AccuTerm will respond:

CSI 0 n

CSI 6 n Request cursor position. AccuTerm will respond:

CSI line ; column R

CSI ? 1 5 n

Request printer status. If a printer is defined for the current configuration, AccuTerm will respond:

CSI ? 1 0 n

If no printer is defined the response will be:

CSI ? 1 3 n

Programming Function Keys

The AccuTerm VT-220 emulation supports the ability to download up to 15 function key values (the shifted function keys **F6-F20**) from the host computer. The following device control sequence is used to program the function keys.

DCS n | key / value ; . . . key / value ST

If **n** is 1, old key definitions are replaced by new definitions; if **n** is 0, the definition of all 15 shifted function keys are cleared before loading any new definitions. The function key to be programmed is specified by **key** according to the ANSI Function Key Table (Appendix C), and the sequence for that key is specified by **value**. **Value** is specified in hexadecimal.

Note: the “Map F11 through F20 ...” check box in the **Keyboard Settings** category, **Settings** dialog, must be checked to access programmed function keys from **F13** to **F20**. When this item is checked, function keys **F11-F20** are mapped onto the control function keys; e.g. **F15** is mapped to **CTRL+F5**, **SHIFT+F13** is mapped to **CTRL+SHIFT+F3**.

VT-52 Escape Sequences

ESC A	Cursor up.
ESC B	Cursor down.
ESC C	Cursor right.
ESC D	Cursor left.
ESC H	Cursor home.
ESC I	Reverse line feed.
ESC J	Erase from cursor to end of screen.
ESC K	Erase from cursor to end of line.
ESC V	Print cursor line.
ESC]	Print screen.
ESC W	Transparent print mode.
ESC X	Cancel transparent print mode.
ESC ^	Auto print mode.
ESC _	Cancel auto print mode.
ESC =	Keypad application mode on.
ESC >	Keypad application mode off.
ESC <	Enter VT220 mode.
ESC Z	Identify terminal.
ESC Y	<i>line col</i> Cursor position.

PICK PC CONSOLE PROGRAMMING

The Pick Operating System on the IBM PC has its own terminal type for the system console (term type I). AccuTerm emulates the Pick PC console thus allowing you to use the same term type codes for all your terminals in a Pick PC environment. The advantage of using this emulation is that it gives you direct access to all the colors and attributes available to the PC.

Operating Modes

ESC * Y Set the video mode to 80 by 25 monochrome mode. This command is ignored by AccuTerm.

ESC *] Sets the video mode to 80 by 25 color. This command is ignored by AccuTerm.

GS If the graphics mode is enabled, this command puts the terminal in the Tektronix emulation mode. If the graphics mode is disabled, this command is ignored.

ESC % ! 0 Enters Tektronix 4100 graphics mode.

Cursor Positioning

BS or ESC * HT

Cursor left. The cursor is moved left one position. If the cursor is at the first column, it is moved to the last column of the previous line.

ESC * DC3 Cursor right. Moves the cursor right one position. If the cursor is at the last column, it will wrap to the first column of the next line.

LF Cursor down. Moves the cursor down one row. If the cursor is on the last row of the screen and the terminal is not in protected mode, the screen scrolls up one line.

ESC * LF Cursor up. Moves the cursor up one row. If the cursor is at the top row and the terminal is not in protected mode, the screen is scrolled down one line.

CR Carriage return. Moves the cursor to the leftmost column of the current row.

ESC * STX Cursor home. Moves the cursor to the upper left corner of the screen.

ESC = c r

This command positions the cursor to column **c** and row **r**. **c** and **r** are the binary cursor positions, numbered from 0 (top row, left column). For example, to position the cursor to column 10 row 10, you would send **ESC = LF LF** where **LF** is the ASCII representation of character 10.

Erasing and Editing

FF or **ESC * SOH**

Clears the screen and moves the cursor to the upper left corner of the screen.

ESC * ETX Clears from the cursor position to the end of the screen.

ESC * EOT Clears from the cursor position to the end of the line.

ESC * h Insert character. A blank is inserted at the current cursor position. All characters from the cursor position to the right are moved right one column.

ESC * i Delete character. The character at the current cursor position is deleted. All characters to the right of the cursor are moved left one column and a blank is placed on the last column of the line.

ESC * j Insert line. A blank line is inserted before the current line. The current line and all lines below are moved down one row.

The bottom line of the screen is lost. The cursor is moved to the left most column of the inserted line.

ESC * k Delete line. The current line is deleted. All lines below the current line are moved up one position. The bottom line is blank. The cursor is moved to the left most column.

Video Attributes

On the PC there are several different ways to set the video attributes. The first is through escape sequences that start and stop individual types like Underline, Reverse, etc. The second is by sending the desired foreground and background colors. One note here, the PC monitor treats the dim attribute as the protected attribute. To change video attributes, you must use the following escape sequences.

ESC * ENQ Start blinking characters.

ESC * ACK Stop blinking characters.

ESC * BEL Start dim (protected) characters.

ESC * BS Stop dim (protected) characters.

ESC * CR Start reverse video characters.

ESC * SO Stop reverse video characters.

ESC * SI Start underlined characters.

ESC * DLE Stop underlined characters.

ESC * n Sets the foreground/background attributes separately where **n** is the code from table below.

<u>Color</u>	<u>Background</u>	Bright <u>Foreground</u>	Dim <u>Foreground</u>
White	!)	9
Yellow	"	*	:
Magenta	#	+	;
Red	\$,	<
Cyan	%	-	=
Green	&	.	>
Blue	'	/	?
Black	(0	@

Protected Attributes

The Pick PC monitor uses the dim attribute as the protected attribute. When the protect mode is enabled, the cursor is not allowed to move to any positions that have protected characters, the auto scroll mode is turned off, and the clear screen functions clear only the non-protected fields.

ESC * VT Turn on the protected mode.

ESC * FF Turn off the protected mode.

ESC * BEL Start protected (dim) characters.

ESC * BS Stop protected (dim) characters.

Printer Control

ESC [? 5 i or **ESC * DC1**

Transparent print mode. Characters are printed, but not displayed. When AccuTerm receives this command, it goes into transparent print mode until it receives the ANSI print off sequence below. This command is useful for printing graphics data to the printer. Since AccuTerm requires the specific ANSI cancel command below to exit transparent print mode, there is less chance of a control character interrupting the graphic printing.

ESC [? 4 i or **ESC * DC2**

Cancel transparent print mode.

APPENDIX A

Wyse Tables

Wyse Cursor Address Table

Row or column	Code	Row or column	Code	Row or column	Code
1	SPACE	33	@	65	`
2	!	34	A	66	a
3	"	35	B	67	b
4	#	36	C	68	c
5	\$	37	D	69	d
6	%	38	E	70	e
7	&	39	F	71	f
8	'	40	G	72	g
9	(41	H	73	h
10)	42	I	74	i
11	*	43	J	75	j
12	+	44	K	76	k
13	,	45	L	77	l
14	-	46	M	78	m
15	.	47	N	79	n
16	/	48	O	80	o
17	0	49	P	81	p
18	1	50	Q	82	q
19	2	51	R	83	r
20	3	52	S	84	s
21	4	53	T	85	t
22	5	54	U	86	u
23	6	55	V	87	v
24	7	56	W	88	w
25	8	57	X	89	x
26	9	58	Y	90	y
27	:	59	Z	91	z
28	;	60	[92	{
29	<	61	\	93	
30	=	62]	94	}
31	>	63	^	95	~
32	?	64	_	96	DEL

Wyse Attribute Code Table

<u>Code</u>	<u>Attribute Type</u>
0	Normal
1	Invisible
2	Blink
3	Invisible
4	Reverse
5	Reverse, invisible
6	Reverse, blink
7	Reverse, invisible
8	Underline
9	Underline, Invisible
:	Underline, blink
;	Underline, blink, invisible
<	Underline, reverse
=	Underline, reverse, invisible
>	Underline, reverse, blink
?	Underline, reverse, blink, invisible
p	Dim
q	Dim, invisible
r	Dim, blink
s	Dim, invisible
t	Dim, reverse
u	Dim, reverse, invisible
v	Dim, reverse, blink
w	Dim, reverse, invisible
x	Dim, underline
y	Dim, underline, invisible
z	Dim, underline, blink
{	Dim, underline, blink, invisible
	Dim, underline, reverse
}	Dim, underline, reverse, invisible
~	Dim, underline, reverse, blink

Wyse Graphic Character Table

<u>Graphic</u>	<u>Code</u>	<u>Graphic</u>	<u>Code</u>
T	0	+	8
L	1	├	9
┐	2	—	:
└	3	░	;
┼	4	=	<
┘	5	⊥	=
	6		>
█	7	▒	?

Wyse Function Key Table

<u>Key</u>	<u>Normal</u> <u>Value</u>	<u>Shifted</u> <u>Value</u>	<u>Normal</u> <u>Field</u>	<u>Shifted</u> <u>Field</u>
F1	@	`	0	P
F2	A	a	1	Q
F3	B	b	2	R
F4	C	c	3	S
F5	D	d	4	T
F6	E	e	5	U
F7	F	f	6	V
F8	G	g	7	W
F9	H	h	8	X
F10	I	i	9	Y
F11	J	j	:	Z
F12	K	k	;	[
F13	L	l	<	\
F14	M	m	=]
F15	N	n	>	^
F16	O	o	?	_
BKSP	"	'		
TAB	!	&		
INS	Q	p		
DEL	5	6		
HOME	*	/		
END	[]		
PGUP	:	;		
PGDN	R	w		
LEFT	-	2		
RIGHT	.	3		
UP	+	0		
DOWN	,	1		
ESC	SPACE	%		
ENTER	\$)		
KPD ENTER	S	4		

Wyse Key Code Table

Key sequences generated when using Wyse 50 & Wyse 60 emulations

PC <u>Key</u>	Terminal <u>Key</u>	<u>Sequence</u>
BKSP	BKSP	<i>BS</i>
SHIFT+BKSP	SHIFT+BKSP	<i>BS</i>
TAB	TAB	<i>HT</i>
SHIFT+TAB	SHIFT+TAB	<i>ESC I</i>
INS	INS	<i>ESC q</i>
SHIFT+INS	SHIFT+INS	<i>ESC r</i>
DEL	DEL	<i>DEL</i>
SHIFT+DEL	DEL LINE	<i>ESC R</i>
HOME	HOME	<i>RS</i>
SHIFT+HOME	SHIFT+HOME	<i>ESC {</i>
END	CLR LINE	<i>ESC T</i>
SHIFT+END	CLR SCRN	<i>ESC Y</i>
PGUP	PGUP	<i>ESC J</i>
SHIFT+PGUP	SHIFT+PGUP	<i>ESC J</i>
PGDN	PGDN	<i>ESC K</i>
SHIFT+PGDN	SHIFT+PGDN	<i>ESC K</i>
LEFT	LEFT	<i>BS</i>
SHIFT+LEFT	SHIFT+LEFT	<i>BS</i>
RIGHT	RIGHT	<i>FF</i>
SHIFT+RIGHT	SHIFT+RIGHT	<i>FF</i>
UP	UP	<i>VT</i>
SHIFT+UP	SHIFT+UP	<i>VT</i>
DOWN	DOWN	<i>LF</i>
SHIFT+DOWN	SHIFT+DOWN	<i>LF</i>
ESC	ESC	<i>ESC</i>
SHIFT+ESC	SHIFT+ESC	<i>ESC</i>
ENTER	RETURN	<i>CR</i>
SHIFT+ENTER	SHIFT+ENTER	<i>CR</i>
KPD ENTER	ENTER	<i>CR</i>
SHIFT+KPD ENTER	SHIFT+ENTER	<i>CR</i>

Wyse Key Code Table (continued)

Key sequences generated when using Wyse 50 & Wyse 60 emulations

F1	F1	<i>SOH @ CR</i>
SHIFT+F1	SHIFT+F1	<i>SOH ` CR</i>
F2	F2	<i>SOH A CR</i>
SHIFT+F2	SHIFT+F2	<i>SOH a CR</i>
F3	F3	<i>SOH B CR</i>
SHIFT+F3	SHIFT+F3	<i>SOH b CR</i>
F4	F4	<i>SOH C CR</i>
SHIFT+F4	SHIFT+F4	<i>SOH c CR</i>
F5	F5	<i>SOH D CR</i>
SHIFT+F5	SHIFT+F5	<i>SOH d CR</i>
F6	F6	<i>SOH E CR</i>
SHIFT+F6	SHIFT+F6	<i>SOH e CR</i>
F7	F7	<i>SOH F CR</i>
SHIFT+F7	SHIFT+F7	<i>SOH f CR</i>
F8	F8	<i>SOH G CR</i>
SHIFT+F8	SHIFT+F8	<i>SOH g CR</i>
F9	F9	<i>SOH H CR</i>
SHIFT+F9	SHIFT+F9	<i>SOH h CR</i>
F10	F10	<i>SOH I CR</i>
SHIFT+F10	SHIFT+F10	<i>SOH i CR</i>
F11	F11	<i>SOH J CR</i>
SHIFT+F11	SHIFT+F11	<i>SOH j CR</i>
F12	F12	<i>SOH K CR</i>
SHIFT+F12	SHIFT+F12	<i>SOH k CR</i>
CTRL+F1	F11	<i>SOH J CR</i>
SHIFT+CTRL+F1	SHIFT+F11	<i>SOH j CR</i>
CTRL+F2	F12	<i>SOH K CR</i>
SHIFT+CTRL+F2	SHIFT+F12	<i>SOH k CR</i>
CTRL+F3	F13	<i>SOH L CR</i>
SHIFT+CTRL+F3	SHIFT+F13	<i>SOH l CR</i>
CTRL+F4	F14	<i>SOH M CR</i>
SHIFT+CTRL+F4	SHIFT+F14	<i>SOH m CR</i>
CTRL+F5	F15	<i>SOH N CR</i>
SHIFT+CTRL+F5	SHIFT+F15	<i>SOH n CR</i>
CTRL+F6	F16	<i>SOH O CR</i>
SHIFT+CTRL+F6	SHIFT+F16	<i>SOH o CR</i>

APPENDIX B

Viewpoint Tables

Viewpoint Cursor Address Table

<u>Column</u>	<u>Code</u>	<u>Column</u>	<u>Code</u>	<u>Column</u>	<u>Code</u>
1	NUL	36	5	71	p
2	SOH	37	6	72	q
3	STX	38	7	73	r
4	ETX	39	8	74	s
5	EOT	40	9	75	t
6	ENQ	41	@	76	u
7	ACK	42	A	77	v
8	BEL	43	B	78	w
9	BS	44	C	79	x
10	HT	45	D	80	y
11	DLE	46	E		
12	DC1	47	F	<u>Row</u>	<u>Code</u>
13	DC2	48	G	1	@
14	DC3	49	H	2	A
15	DC4	50	I	3	B
16	NAK	51	P	4	C
17	SYN	52	Q	5	D
18	ETB	53	R	6	E
19	CAN	54	S	7	F
20	EM	55	T	8	G
21	SPACE	56	U	9	H
22	!	57	V	10	I
23	"	58	W	11	J
24	#	59	X	12	K
25	\$	60	Y	13	L
26	%	61	\	14	M
27	&	62	a	15	N
28	'	63	b	16	O
29	(64	c	17	P
30)	65	d	18	Q
31	0	66	e	19	R
32	1	67	f	20	S
33	2	68	g	21	T
34	3	69	h	22	U
35	4	70	i	23	V
				24	W

Viewpoint Attribute Code Table

<u>Code</u>	<u>Attribute Type</u>
@	Normal
A	Dim
B	Normal Blinking
C	Dim Blinking
P	Reverse
Q	Dim Reverse
R	Reverse Blinking
S	Dim Reverse Blinking
`	Underlined
A	Dim Underlined
B	Underlined Blinking
C	Dim Underline Blinking
D	Invisible

Viewpoint 60 Graphic Character Table

<u>Graphic</u>	<u>Code</u>
┘	@ or A or B or C
└	D or E or F or G
┐	H or I or J or K
┌	L or M or N or O
└┘	P or Q or R or S
┐┌	T or U or V or W
└┐	X or Y or Z or [
┘┘	\ or] or ^ or _
└└	` or a or b or c
┐┐	d or e or f or g
└┐┘	h or i or j or k

Viewpoint Key Code Table

Key sequences generated when using ADDS Viewpoint A2, Viewpoint
Enhanced, Viewpoint 60 and Procomm VP60 emulations

PC <u>Key</u>	Terminal <u>Key</u>	<u>Sequence</u>
BKSP	BKSP	<i>BS</i>
SHIFT+BKSP	SHIFT+BKSP	<i>BS</i>
TAB	TAB	<i>HT</i>
SHIFT+TAB	SHIFT+TAB	<i>ESC O</i>
INS	INS	<i>ESC q</i>
SHIFT+INS	SHIFT+INS	<i>ESC r</i>
DEL	DEL	<i>DEL</i>
SHIFT+DEL	DEL LINE	<i>ESC 7</i>
HOME	HOME	<i>SOH</i>
SHIFT+HOME	SHIFT+HOME	<i>SOH</i>
END	CLR LINE	<i>ESC K</i>
SHIFT+END	CLR SCRN	<i>ESC k</i>
PGUP	PGUP	<i>ESC J</i>
SHIFT+PGUP	SHIFT+PGUP	<i>ESC J</i>
PGDN	PGDN	<i>ESC </i>
SHIFT+PGDN	SHIFT+PGDN	<i>ESC </i>
LEFT	LEFT	<i>NAK</i>
SHIFT+LEFT	SHIFT+LEFT	<i>NAK</i>
RIGHT	RIGHT	<i>ACK</i>
SHIFT+RIGHT	SHIFT+RIGHT	<i>ACK</i>
UP	UP	<i>SUB</i>
SHIFT+UP	SHIFT+UP	<i>SUB</i>
DOWN	DOWN	<i>LF</i>
SHIFT+DOWN	SHIFT+DOWN	<i>LF</i>
ESC	ESC	<i>ESC</i>
SHIFT+ESC	SHIFT+ESC	<i>ESC</i>
ENTER	RETURN	<i>CR</i>
SHIFT+ENTER	SHIFT+ENTER	<i>CR</i>
KPD ENTER	ENTER	<i>CR</i>
SHIFT+KPD ENTER	SHIFT+ENTER	<i>CR</i>

Viewpoint Key Code Table (continued)

PC <u>Key</u>	Terminal <u>Key</u>	<u>Sequence</u>
F1	F1	STX 1 CR
SHIFT+F1	SHIFT+F1	STX ! CR
F2	F2	STX 2 CR
SHIFT+F2	SHIFT+F2	STX " CR
F3	F3	STX 3 CR
SHIFT+F3	SHIFT+F3	STX # CR
F4	F4	STX 4 CR
SHIFT+F4	SHIFT+F4	STX \$ CR
F5	F5	STX 5 CR
SHIFT+F5	SHIFT+F5	STX % CR
F6	F6	STX 6 CR
SHIFT+F6	SHIFT+F6	STX & CR
F7	F7	STX 7 CR
SHIFT+F7	SHIFT+F7	STX ' CR
F8	F8	STX 8 CR
SHIFT+F8	SHIFT+F8	STX (CR
F9	F9	STX 9 CR
SHIFT+F9	SHIFT+F9	STX) CR
F10	F10	STX : CR
SHIFT+F10	SHIFT+F10	STX * CR
F11	F11	STX ; CR
SHIFT+F11	SHIFT+F11	STX + CR
F12	F12	STX < CR
SHIFT+F12	SHIFT+F12	STX , CR
CTRL+F1	F11	STX ; CR
SHIFT+CTRL+F1	SHIFT+F11	STX + CR
CTRL+F2	F12	STX < CR
SHIFT+CTRL+F2	SHIFT+F12	STX , CR
CTRL+F3	F13	STX = CR
SHIFT+CTRL+F3	SHIFT+F13	STX - CR
CTRL+F4	F14	STX > CR
SHIFT+CTRL+F4	SHIFT+F14	STX . CR
CTRL+F5	F15	STX ? CR
SHIFT+CTRL+F5	SHIFT+F15	STX / CR
CTRL+F6	F16	STX 0 CR
SHIFT+CTRL+F6	SHIFT+F16	STX SPACE CR

APPENDIX C

ANSI Tables

ANSI Attribute Code Table

<u>Code</u>	<u>Attribute Type</u>
0	Normal
1	Bold
2	Dim
4	Underline
5	Blinking
7	Reverse
8	Blank
22	Dim & Bold off
24	Underline off
25	Blinking off
27	Reverse off
28	Blank off
30	Black character
31	Red character
32	Green character
33	Yellow character
34	Blue character
35	Magenta character
36	Cyan character
37	White character
40	Black background
41	Red background
42	Green background
43	Yellow background
44	Blue background
45	Magenta background
46	Cyan background
47	White background

ANSI Function Key Table

Key to <u>Program</u>	<u>Key</u>
F6	17
F7	18
F8	19
F9	20
F10	21
F11	23
F12	24
F13	25
F14	26
F15	28
F16	29
F17	31
F18	32
F19	33
F20	34

ANSI Key Code Table

Key sequences generated when using VT-220 or ANSI emulations

PC <u>Key</u>	Terminal <u>Key</u>	<u>Sequence</u>
BKSP	BKSP	BS
SHIFT+BKSP	SHIFT+BKSP	BS
TAB	TAB	HT
SHIFT+TAB	SHIFT+TAB	CSI Z
INS		
SHIFT+INS		
DEL	DEL	DEL
SHIFT+DEL	DEL LINE	DEL
HOME	HOME	CSI H
SHIFT+HOME	SHIFT+HOME	CSI H
END		
SHIFT+END		
PGUP	PGUP	CSI V
SHIFT+PGUP	SHIFT+PGUP	CSI V
PGDN	PGDN	CSI U
SHIFT+PGDN	SHIFT+PGDN	CSI U
LEFT	LEFT	CSI D
SHIFT+LEFT	SHIFT+LEFT	CSI D
RIGHT	RIGHT	CSI C
SHIFT+RIGHT	SHIFT+RIGHT	CSI C
UP	UP	CSI A
SHIFT+UP	SHIFT+UP	CSI A
DOWN	DOWN	CSI B
SHIFT+DOWN	SHIFT+DOWN	CSI B
ESC	ESC	ESC
SHIFT+ESC	SHIFT+ESC	ESC
ENTER	RETURN	CR
SHIFT+ENTER	SHIFT+ENTER	CR
KPD ENTER	ENTER	CR
SHIFT+KPD ENTER	SHIFT+ENTER	CR

When using 7 bit controls, **CSI** is sent as **ESC [** and **SS3** is sent as **ESC O**.

ANSI Key Code Table (continued)

PC <u>Key</u>	Terminal <u>Key</u>	<u>Sequence</u>
F1	PF1	SS3 P
F2	PF2	SS3 Q
F3	PF3	SS3 R
F4	PF4	SS3 S
F5		CSI M
F6	F6	CSI 1 7 ~
F7	F7	CSI 1 8 ~
F8	F8	CSI 1 9 ~
F9	F9	CSI 2 0 ~
F10	F10	CSI 2 1 ~
F11	F11	CSI 2 3 ~
F12	F12	CSI 2 4 ~
CTRL+F1	F11	CSI 2 3 ~
CTRL+F2	F12	CSI 2 4 ~
CTRL+F3	F13	CSI 2 5 ~
CTRL+F4	F14	CSI 2 6 ~
CTRL+F5	F15	CSI 2 8 ~
CTRL+F6	F16	CSI 2 9 ~
CTRL+F7	F17	CSI 3 1 ~
CTRL+F8	F18	CSI 3 2 ~
CTRL+F9	F19	CSI 3 3 ~
CTRL+F10	F20	CSI 3 4 ~

When using 7 bit controls, **CSI** is sent as **ESC** [and **SS3** is sent as **ESC** O.

APPENDIX D

ASCII Codes

<u>ASCII</u>	<u>Decimal</u>	<u>Hex</u>	<u>ASCII</u>	<u>Decimal</u>	<u>Hex</u>
NUL	0	00	SPACE	32	20
SOH	1	01	!	33	21
STX	2	02	"	34	22
ETX	3	03	#	35	23
EOT	4	04	\$	36	24
ENQ	5	05	%	37	25
ACK	6	06	&	38	26
BEL	7	07	'	39	27
BS	8	08	(40	28
HT	9	09)	41	29
LF	10	0A	*	42	2A
VT	11	0B	+	43	2B
FF	12	0C	,	44	2C
CR	13	0D	-	45	2D
SO	14	0E	.	46	2E
SI	15	0F	/	47	2F
DLE	16	10	0	48	30
DC1	17	11	1	49	31
DC2	18	12	2	50	32
DC3	19	13	3	51	33
DC4	20	14	4	52	34
NAK	21	15	5	53	35
SYN	22	16	6	54	36
ETB	23	17	7	55	37
CAN	24	18	8	56	38
EM	25	19	9	57	39
SUB	26	1A	:	58	3A
ESC	27	1B	;	59	3B
FS	28	1C	<	60	3C
GS	29	1D	=	61	3D
RS	30	1E	>	62	3E
US	31	1F	?	63	3F

ASCII Codes (continued)

@	64	40	`	96	60
A	65	41	a	97	61
B	66	42	b	98	62
C	67	43	c	99	63
D	68	44	d	100	64
E	69	45	e	101	65
F	70	46	f	102	66
G	71	47	g	103	67
H	72	48	h	104	68
I	73	49	i	105	69
J	74	4A	j	106	6A
K	75	4B	k	107	6B
L	76	4C	l	108	6C
M	77	4D	m	109	6D
N	78	4E	n	110	6E
O	79	4F	o	111	6F
P	80	50	p	112	70
Q	81	51	q	113	71
R	82	52	r	114	72
S	83	53	s	115	73
T	84	54	t	116	74
U	85	55	u	117	75
V	86	56	v	118	76
W	87	57	w	119	77
X	88	58	x	120	78
Y	89	59	y	121	79
Z	90	5A	z	122	7A
[91	5B	{	123	7B
\	92	5C		124	7C
]	93	5D	}	125	7D
^	94	5E	~	126	7E
_	95	5F	DEL	127	7F

ASCII Codes (continued)

	128	80	<i>DCS</i>	144	90
	129	81	<i>PU1</i>	145	91
	130	82	<i>PU2</i>	146	92
	131	83	<i>STS</i>	147	93
<i>IND</i>	132	84	<i>CCH</i>	148	94
<i>NEL</i>	133	85	<i>MW</i>	149	95
<i>SSA</i>	134	86	<i>SPA</i>	150	96
<i>ESA</i>	135	87	<i>EPA</i>	151	97
<i>HTS</i>	136	88		152	98
<i>HTJ</i>	137	89		153	99
<i>VTS</i>	138	8A		154	9A
<i>PLD</i>	139	8B	<i>CSI</i>	155	9B
<i>PLU</i>	140	8C	<i>ST</i>	156	9C
<i>RI</i>	141	8D	<i>OSC</i>	157	9D
<i>SS2</i>	142	8E	<i>PM</i>	158	9E
<i>SS3</i>	144	8F	<i>APC</i>	159	9F

APPENDIX E

Pick File Transfer and Server Programs

Included with AccuTerm is a set of Pick/BASIC programs which may be used to send or receive data from a Pick host machine. The programs provide three TCL level file transfer utilities (FT, FTD and FTPICK), and two callable BASIC subroutines (FTS and DOSSVC) which may be used to integrate Windows and DOS programs and file transfer functions into new or existing Pick applications. A standalone Kermit utility is also provided.

Besides the file transfer utilities include with AccuTerm, is a "server" program, which, when running, places the Pick host in a "server" mode. When in this mode, other Windows applications can retrieve and update host files using the ActiveX Server object which AccuTerm makes available (see **AccuTerm Object Reference**). *Note: this server is also available via a DDE connection.*

In order to understand the file transfer facilities provided by AccuTerm, the differences between the DOS and Pick file systems must be examined. Traditional DOS files may be programs, documents or data files; in the typical Pick system, programs and documents are usually stored as individual items (records) in a file, whereas a data file is collection of items (records) in a file.

When performing file transfer operations between DOS and Pick, it is important to realize the difference between "document files" and "data files".

When transferring "document files" between DOS and Pick, use the **FT** utility. This utility transfers Pick items to DOS files and vice versa.

When transferring "data files" between DOS and Pick, use the **FTD** utility. This utility transfers a group of Pick items to a single DOS file and vice versa.

The **FTPICK** utility is used to transfer data from one Pick machine to another, either directly, or using an intermediate DOS file.

AccuTerm's file transfer utilities support two transfer protocols. The ASCII protocol may be used for simple text file transfer. When using the ASCII protocol, the sending machine simply "prints" the file while the receiving machine "inputs" data and records it in a file. There is no synchronization between the two machines, therefore time delays are used to prevent the sender from overrunning the receiver. The ASCII protocol uses the ASCII **SUB** (control-Z) control code to indicate end-of-file.

Kermit is the other file transfer protocol supported by the AccuTerm file transfer utilities. The Kermit protocol may be used to send any type of data, and sender and receiver negotiate certain characteristics prior to performing the transfer. The two machines are synchronized. Data is transferred in packets, and each packet is checked for errors. If errors are detected, the packet is re-transmitted. The Kermit protocol uses the ASCII **SOH** (control-A) control code for a packet marker.

Kermit is the recommended file transfer protocol because of the automatic error detection and correction. Also, even with the increased overhead ensuing from Kermit, because the sender and receiver are synchronized and no time delays are required, the Kermit protocol usually performs better than the ASCII protocol when receiving a DOS file.

Installing the Pick File Transfer Programs


To install the PICK/BASIC file transfer programs onto the PICK host, perform the following steps. *Before installing the file transfer programs, please see platform specific notes at the end of this section.*

- 1) Create a file for the PICK/BASIC programs:

```
CREATE-FILE FTBP 1 3
```

Note: some Pick and Pick like systems block control characters (CIE/GA/Icon) or alter the case of alpha characters (UniVerse). These features must be disabled to use and install the file transfer programs.

- 2) Set the ASCII inter-line delay to at least 350ms (from the main menu, select **T**ools **S**ettings, then click on the **F**ile **T**ransfer tab).

- 3) With the PICK host at command level (TCL), select **Transfer Upload** from the main menu or click the  button.
- 4) In the **AccuTerm File Upload** panel, click the **Browse** button and select the directory where the Pick file transfer programs are installed (usually \ATWIN\PICKBP).
- 5) Select the file “ftpinst.txt” (*or other .TXT file as specified in the platform specific instructions at the end of this section*) in the file selection list.
- 6) Select **ASCII** protocol, then press **ALT + U** or click the **Upload** button. When the transfer is complete (about 2 minutes), press the **ENTER** key to close the file transfer status window. Press the **ENTER** key once or twice to clear out the input buffer.
- 7) Type **FTPINST** at command level (TCL). The file transfer installation program will be compiled and run.
- 8) Enter the directory where the Pick file transfer programs reside (same path as in step 4 above). If the path is C:\ATWIN\PICKBP, simply press **ENTER**.
- 9) The file transfer installation program will then transfer, compile and catalog the remainder of the Pick/BASIC file transfer programs.
- 10) If another file transfer utility (such as ViaDuct) previously installed has used the same names as the AccuTerm FTBP programs, you may have to remove those entries from your MD or VOC, then re-catalog the programs in the FTBP file.

Using the FT Utility

The FT utility will transfer specified (or selected) Pick items to (or from) DOS files. To use the FT utility, enter FT at TCL, then answer the prompts. On-line help is provided for each prompt.

Sending Pick Items to DOS Files

Use the FT utility to send Pick items to DOS files. When prompted:

```
(S)end, (R)eceive, (C)onfigure, (O)ptions,  
(H)elp or (E)xit ?
```

Answer S to send, then select the file transfer protocol:

```
File transfer protocol: (A)SCII or (K)ermit ?
```

Kermit is recommended.

```
Enter source (Pick) file name:
```

Enter the name of the Pick file which contains the items to be sent.

```
Enter source (Pick) item list:
```

Enter the Pick item-IDs to be sent. If you do not enter item-IDs, any previously selected (SELECT, GET-LIST, etc.) items will be transferred; if you enter an asterisk (*), all items in the file will be transferred. To specify a saved list, enter an open parenthesis followed by the list name. Separate item-IDs with blanks.

```
Enter target (DOS) directory name:
```

Enter the DOS directory name including drive letter if desired. Separate directory names in a full path using back-slashes.

```
Enter target (DOS) file name:
```

Enter the DOS file name, including extension. If multiple items are being sent, multiple file names may be specified here. Each Pick item will be written to a separate DOS file. If no file names are specified here, the Pick item-id will be used as the DOS file name, subject to the normal DOS file name restrictions.

Sending DOS Files to Pick Items

Use the FT utility to send DOS files to Pick items. When prompted:

```
(S)end, (R)eceive, (C)onfigure, (O)ptions,  
(H)elp or (E)xit ?
```

Answer **R** to receive, then select the file transfer protocol:

```
File transfer protocol: (A)SCII or (K)ermit ?
```

Kermit is recommended.

```
Enter source (DOS) directory name:
```

Enter the name of the DOS directory which contains the files to be received.

```
Enter source (DOS) file name:
```

Enter the name(s) of the DOS file(s) to be received. Separate multiple file names with blanks. Wildcard characters (?) and (*) are valid.

```
Enter target (Pick) file name:
```

Enter the Pick file name where the received DOS files are to be saved. Each received DOS file is stored as a Pick item in this file.

```
Enter target (Pick) item list:
```

Enter the Pick item-ID(s) for the received DOS file(s). If multiple files are being received, multiple item-IDs may be specified here. If no item-IDs are specified here, the DOS file name will be used as the Pick item-ID.

FT Options and Settings

Several configuration (permanent) and option (temporary) settings are available for the FT utility. To change configuration settings, enter C in response to the first prompt. A list of configuration settings will be displayed:

```
Current parameter settings are:
1. Transfer mode      Text
2. Preserve file extension      Yes
3. Overwrite existing item      Yes
4. Large item treatment Truncate
```

Enter parameter number (1-4) to modify:

The setting for any of these parameters may be changed by entering the parameter number. Possible transfer modes are text and binary. Text causes attribute mark characters to be translated to carriage return / line feeds and vice versa. Large item treatment may be set to truncate items at about 30000 bytes, split large items into multiple items of about 30000 bytes each, or no action (for those systems with no item size limitation). Split items use a special item-id generated by concatenating the original item-id, an underscore (_), and a sequential number starting with 001.

To change option settings, enter O in response to the first prompt. A list of option settings will be displayed. Note that the default value of the options is stored in the configuration settings:

```
Current parameter settings are:
1. Transfer mode      Text
2. Preserve file extension      Yes
```

An alternate way to set the value of a configuration or option setting is to answer the prompt with the parameter number, an equal sign, and the first letter of the value to which the parameter is to be set. For example, to set the first configuration parameter to "binary", respond to "Enter parameter..." with 1=B. This alternate method is useful in cases where the FT utility is being "driven" by a PROC or by another program using the EXECUTE statement.

Using the FTD Utility

The FTD utility will transfer a set of specified (or selected) Pick items to (or from) a DOS file. The fields (attributes) to transfer may be specified, and an optional header record may be used.

The default file format used by the FTD utility is ASCII delimited text. In this format, DOS records are separated by a carriage return / line feed and fields are separated by tabs or commas. Fields which contain an embedded field delimiter are enclosed in quotation marks. Embedded quotation marks are duplicated. This file format is easily imported to or exported from many DOS or Windows database or spreadsheet programs.

Certain file extensions are recognized by AccuTerm, and cause the imported / exported data to be automatically converted to/from a pre-defined file format. Default extensions which perform the automatic conversion are:

.db2	dBase II
.dbf	dBase III/IV
.sym	Symphony
.wb1	Quattro
.wk1	Lotus v1-v2
.wks	Lotus v3-v5
.xls	Excel

Certain limitations apply when using the auto-convert feature. There is a maximum field length of 255 characters, maximum record length of 16384 characters and a maximum of 1024 fields per record. When converting to a spreadsheet file, each column is considered a field and each row is a record. Also, the conversion process may remove leading zeros from numbers, and leading blanks from strings.

Note: you can customize the file extensions which AccuTerm recognizes, and instruct AccuTerm to perform the automatic conversion to/from one of the supported file formats. To modify the file types, use Windows Explorer, and find the "atedtype.exe" file in the AccuTerm main program folder. Double-click on this file name to start the configuration program.

When accessing Excel workbook files, the FTD utility has the ability to reference explicit worksheets and ranges of cells. To use this feature, specify the Excel file name as:

```
d:\path\file.xls!sheet!range
```

where `d:\path\file.xls` is the path of the Excel file, `sheet` is the worksheet name or tab number (`sheet` may be null to use the first worksheet in the workbook), `range` is the range of cells to access. Either a named range may be used or a cell location (e.g. A4) or cell range (e.g. B3:F42) may be specified.

When sending data from Pick to Excel, the range identifies the upper-left cell for the destination of the download; the number of columns and rows is determined by the Pick data being sent.

When sending data from Excel to Pick, the range specifies the entire block of data to send. If want to specify the upper-left corner of the block, follow the cell with a trailing colon (e.g. B3:).

To use the FTD utility, enter FTD at TCL, then answer the prompts. On-line help is provided for each prompt.

Sending a Pick File to a DOS File

Use the FTD utility to send a Pick file to a DOS file. When prompted:

```
(S)end, (R)eceive, (C)onfigure, (O)ptions,  
(H)elp or (E)xit ?
```

Answer S to send, then select the file transfer protocol:

```
File transfer protocol: (A)SCII or (K)ermit ?
```

Kermit is recommended.

```
Enter source (Pick) file name:
```

Enter the name of the Pick file which contains the items to be sent.

```
Enter source (Pick) item list:
```

Enter the Pick item-IDs to be sent. If you do not enter item-IDs, any previously selected (SELECT, GET-LIST, etc.) items will be transferred; if you enter an asterisk (*), all items in the file will be transferred. To specify a saved list, enter an open parenthesis followed by the list name. Separate item-IDs with blanks.

```
Enter target (DOS) file (path) name:
```

Enter the full DOS file name including drive letter and directory, if desired. Separate directory names in a full path using back-slashes.

```
Attributes to transfer:
```

Enter the list of attributes to be transferred from each item. Enter either the attribute number or attribute name as defined in the file dictionary. An asterisk (*) may be entered here to transfer all attributes including the Item-ID, from all selected items. A plus-sign (+) may be entered to transfer all attributes *except* the Item-ID. Attribute definitions containing A or F correlatives cannot be specified.

```
Generate Header Record (<Y>/N):
```

Answer Y if you want the first record in the DOS file to be a "header record". The header record specifies the names of the attributes specified above, and may be used as column or field identifiers by certain DOS

applications. If an asterisk was entered for the attribute list, the header record will contain only an asterisk.

Explode MultiValue Fields (Y/<N>):

If asterisk was entered for the attribute list, the preceding prompt will be bypassed. Otherwise, answer Y to explode all multi-valued fields. When multi-valued fields are exploded, a separate data record will be generated for each multi-value, with any single-valued attributes repeated for each multi-value.

Sending a DOS File to a Pick File

Use the FTD utility to send a DOS file to a Pick file. When prompted:

(S)end, (R)eceive, (C)onfigure, (O)ptions,
(H)elp or (E)xit ?

Answer R to receive, then select the file transfer protocol:

File transfer protocol: (A)SCII or (K)ermit ?

Kermit is recommended.

Enter source (DOS) file (path) name:

Enter the full DOS file name including drive letter and directory, if desired. Separate directory names in a full path using back-slashes.

Use Header Record (<Y>/N):

Answer Y if you want to use the first record in the DOS file as a "header record". The header record specifies the names (or numbers) of the attributes to be received. If the header record consists of a single asterisk, all attributes in the item are assumed. If you answer Y to this prompt, the next prompt will be "Enter target (Pick) file name", the others will be bypassed.

Attributes to transfer:

If you are using a header record, the preceding prompt will be bypassed. Otherwise, enter the list of attributes to be transferred to each item. Enter

either the attribute number or attribute name as defined in the file dictionary. An asterisk may be entered here to transfer all attributes (entire item). Attribute definitions containing A or F correlatives cannot be specified.

Number of header records to skip:

Enter the number of records at the beginning of the DOS file which are to be considered "header records", and which are to be ignored when receiving the DOS file.

Automatically generate target (Pick) IDs
(Y/<N>):

Answer Y to generate Pick item-IDs for each record. If the item-ID is not specified in response to the "Attributes to transfer" prompt, you must generate item-IDs. The generated item-ID consists of a constant (possibly null) prefix, followed by a sequential number. If item-IDs are not generated, the next two prompts will be skipped.

Auto ID Prefix:

Enter the constant item-ID prefix. Null is acceptable if the generated item-ID is simply a sequential number.

Auto ID Starting Number:

Enter the starting value for automatically generated item-IDs.

Enter target (Pick) file name:

Enter the name of the Pick file where the received data is to be stored.

FTD Options and Settings

Several configuration (permanent) and option (temporary) settings are available for the FTD utility. To change configuration settings, enter C in response to the first prompt. A list of configuration settings will be displayed:

```
Current parameter settings are:
    1. Preserve file extension      Yes
    2. Overwrite existing item     No
    3. Large item treatment        No action
    4. Field delimiter             Tab
```

Enter parameter number (1-4) to modify:

The setting for any of these parameters may be changed by entering the parameter number. The field delimiter (DOS file) may be set to either "Tab" or "Comma".

To change option settings, enter O in response to the first prompt. A list of option settings will be displayed. Note that the default value of the options is stored in the configuration settings:

```
Current parameter settings are:
    1. Preserve file extension      Yes
    2. Overwrite existing item     No
    3. Field delimiter             Tab
```

An alternate way to set the value of a configuration or option setting is to answer the prompt with the parameter number, an equal sign, and the first letter of the value to which the parameter is to be set. For example, to set the third configuration parameter to "tab", respond to "Enter parameter..." with 3=T. This alternate method is useful in cases where the FTD utility is being "driven" by a PROC or by another program using the EXECUTE statement.

Using the FTPICK Utility

The FTPICK utility will transfer specified (or selected) Pick items from one Pick machine to another. To use the FTPICK utility, enter **FTPICK** at **TCL**, then answer the prompts. On-line help is provided for each prompt.

Direct Pick-to-Pick (one step) transfer

The direct Pick-to-Pick transfer requires two sessions running on AccuTerm - one session connected to the file source (sending machine), and one session connected to the target (receiving machine). If you cannot run two sessions, you must use an intermediate DOS file to store the data temporarily (see next section).

To use the FTPICK direct transfer method, activate the AccuTerm session connected to the target machine, and start FTPICK. When prompted:

```
(S)end, (R)eceive, (C)onfigure, (O)ptions,  
(H)elp or (E)xit ?
```

Answer **R** to receive files, then select the direct mode:

```
Receive (D)irectly from PICK file or from  
(I)ntermediate DOS file ?
```

by answering **D**. A panel titled “AccuTerm Pick-to-Pick File Transfer” will be displayed on the target session’s screen. You can cancel the Pick-to-Pick transfer at any time by pressing the **ESC** key while this session is active.

Next, activate the AccuTerm session connected to the source, or sending, machine, and start FTPICK. When prompted:

```
(S)end, (R)eceive, (C)onfigure, (O)ptions,  
(H)elp or (E)xit ?
```

answer **S** to send files. Then enter the name of the Pick file to be sent:

```
Enter PICK source file name:
```

Enter the name of the Pick file which contains the items to be sent.

```
Enter source (Pick) item list:
```

Enter the Pick item-IDs to be sent. If you do not enter item-IDs, any previously selected (**SELECT**, **GET-LIST**, etc.) items will be transferred; if

you enter an asterisk (*), all items in the file will be transferred. To specify a saved list, enter an open parenthesis followed by the list name. Separate item-IDs with blanks.

```
Send (D)irectly to PICK file or to
(I)ntermediate DOS file ?
```

Answer **D** to send the file directly to the target Pick machine.

```
Enter target PICK file name:
```

Enter the file name on the target Pick machine into which the selected items will be written.

Once the target file name has been entered, the receiving session will be activated and the transfer progress will be displayed in the panel. When the transfer is complete, press the **ESC** key to terminate the receiving FTPICK program, or activate the sending session to send another file.

Pick-to-Pick transfer using Intermediate DOS file (two step)

If you cannot open one session to each Pick machine, or if you want to send the same file to several Pick machines, you can save the data in an intermediate DOS file. The procedure is similar to the direct method described above, except that the send must be done first, and a DOS file must be specified for both the send and receive steps.

Step 1: Transferring Pick items to an intermediate DOS file

To use the FTPICK indirect transfer method, open an AccuTerm session connected to the source machine, and start FTPICK. When prompted:

```
(S)end, (R)eceive, (C)onfigure, (O)ptions,
(H)elp or (E)xit ?
```

answer **S** to send files. Then enter the name of the Pick file to be sent:

```
Enter PICK source file name:
```

Enter the name of the Pick file which contains the items to be sent.

Enter source (Pick) item list:

Enter the Pick item-IDs to be sent. If you do not enter item-IDs, any previously selected (SELECT, GET-LIST, etc.) items will be transferred; if you enter an asterisk (*), all items in the file will be transferred. To specify a saved list, enter an open parenthesis followed by the list name. Separate item-IDs with blanks.

Send (D)irectly to PICK file or to
(I)ntermediate DOS file ?

Answer I to send the file to an intermediate DOS file.

Enter intermediate DOS file name:

Enter the full DOS file name of the intermediate file including drive letter and directory, if desired. Separate directory names in a full path using backslashes.

Upon entering the DOS file name, the selected items will be transferred to the specified DOS file in a special format used by the FTPICK utility.

Enter the DOS file name, including extension. If multiple items are being sent, multiple file names may be specified here. Each Pick item will be written to a separate DOS file. If no file names are specified here, the Pick item-id will be used as the DOS file name, subject to the normal DOS file name restrictions.

Step 2: Transferring Pick items from an intermediate DOS file

To use the FTPICK utility to retrieve Pick items from an intermediate DOS file (created by FTPICK), open an AccuTerm session connected to the target machine, and start FTPICK. When prompted:

(S)end, (R)eceive, (C)onfigure, (O)ptions,
(H)elp or (E)xit ?

answer R to receive files. Then enter the name of the intermediate DOS file which contains the desired items:

Receive (D)irectly from PICK file or from
(I)ntermediate DOS file ?

Answer I to retrieve data from an intermediate DOS file.

Enter intermediate DOS file name:

Enter the full DOS file name of the intermediate file including drive letter and directory, if desired. Separate directory names in a full path using backslashes.

Enter target PICK file name:

Enter the file name on the target Pick machine into which the retrieved items will be written.

If the intermediate DOS file is in the correct format, the items contained in the DOS file will be transferred and stored into the target Pick file.

FTPICK Options and Settings

One configuration setting (permanent) and option setting (temporary) is available for the FTPICK utility. To change configuration setting, enter C in response to the first prompt. A list of configuration settings will be displayed:

```
Current parameter settings are:
      1. Overwrite existing item      Yes
```

Enter parameter number (1-1) to modify:

The “overwrite existing file” setting may be changed by entering the parameter number (1).

When using the direct mode, both the sender and receiver must have “Overwrite existing file” set to “Yes” in order for existing items to be overwritten. If either the sender or receiver has this settings set to “No”, existing items will not be overwritten.

To change option setting, enter O in response to the first prompt. A list of option settings will be displayed. Note that the default value of the options is stored in the configuration settings:

```
Current parameter settings are:
      1. Overwrite existing item      Yes
```

Enter parameter number (1-1) to modify:

An alternate way to set the value of a configuration or option setting is to answer the prompt with the parameter number, an equal sign, and the first letter of the value to which the parameter is to be set. For example, to set the first configuration parameter to "binary", respond to "Enter parameter..." with 1=Y. This alternate method is useful in cases where the FTPICK utility is being "driven" by a PROC or by another program using the EXECUTE statement.

Pick Server

When AccuTerm is connected to a Pick host, and the host is running a special server program, **FTSERVER**, included with AccuTerm, other Windows applications (client applications) can request and update data from the Pick host database using the ActiveX Server object.

The `Server` object, in conjunction with the **FTSERVER** host program provides a simple method for performing common host database operations, including reading and writing files, executing commands, calling Pick/BASIC subroutines and converting data. The `Server` object can be used in any ActiveX enabled client application, such as Microsoft Word, Excel, and Visual Basic.

When an AccuTerm session is in server mode, it is dedicated to providing Pick services to client programs, and cannot be used as a normal terminal. A “Server Mode” message is displayed while the session is in server mode. When server mode is terminated, normal terminal functions are resumed. To start server mode, enter “FTSERVER” on the Pick TCL command line. To terminate server mode, click the **Cancel** button.

Configuring the Pick Server

After installing the file transfer programs, use the **FTSETUP** utility to configure the Pick server. You must use **FTSETUP** before you can use the server because the server name needs to be established. You can also enable or disable individual services (read, write, convert, execute) using **FTSETUP**.

Accessing the Pick Host

To access the Pick host, open an AccuTerm session to the desired host, and type **FTSERVER** at the TCL command prompt. AccuTerm should respond by displaying a “Server Mode” status panel identifying the Pick host server name.

Next, in your client application (Word, Excel, VB, etc.) macro environment (Tools ⇒ Macro ⇒ Visual Basic Editor), add a reference to “AccuTerm Pick Server” (Tools ⇒ References). *Note: it is not strictly necessary to add the reference, but doing so allows for automatic parameter type checking and provides a slight performance increase. If you do not add the reference, declare you server object variable as type **Object** rather than as **atPickServer.Server**.* Next, add a module (Insert ⇒ Module). In your Sub or Function, declare a variable for the Pick server object:

```
Dim PickServerObject As atPickServer.Server
```

Next, use the `CreateObject()` function to create an instance of the server object:

```
Set PickServerObject =  
    CreateObject("atPickServer.Server")
```

Use the `Connect` method of the server object to establish a connection to the host:

```
If PickServerObject.Connect() Then ...
```

The `Connect` method returns `True` if the connection is successful.

Use the `ReadItem`, `WriteItem`, `AddItem`, `DeleteItem`, `UnlockItem`, `OConv`, `IConv`, `Execute`, `CallSub`, `Download`, `Upload`, `Export` or `Import` methods to access the host database. Each of the methods is described in detail in the next section.

When you are finished using the server, you can disconnect and destroy the server object:

```
PickServerObject.Disconnect  
Set PickServerObject = Nothing
```

If you are going to use the server object multiple times (perhaps from multiple Subs or Functions), you can make the server object variable global (declare it before all Subs or Functions in the module). Then you could write a common function to create the server object and connect to the host:

```

Dim PickServerObject As atPickServer.Server

Private Function ConnectServer() As Boolean
    On Error Resume Next
    If Not PickServerObject Is Nothing Then
        ConnectServer = True
    Else
        Set PickServerObject =
        CreateObject("atPickServer.Server")
        If Not PickServerObject Is Nothing Then
            If PickServerObject.Connect() Then
                ConnectServer = True
                Exit Function
            Else
                Set PickServerObject = Nothing
            End If
        End If
        ConnectServer = False
    End If
End Function

```

All other Subs or Functions which require host database access would first call the ConnectServer() function to make sure that services are available:

```

Public Function PickRead(File As String, ID As
String) As String
    If ConnectServer() Then
        PickRead = PickServerObject.ReadItem(File,
ID)
    End If
End Function

```

After calling any of the server methods, check the result of the operation by examining the LastError property. To get a description of the error, use the LastErrorMessage property.

```

If PickServerObject.LastError Then
    MsgBox "Server Error: " &
PickServerObject.LastErrorMessage
End If

```


Some standard errors returned by the FTSERVER host program are shown below; other codes may be returned from called Pick/BASIC subroutines.

-1	End of file
201	Unable to open file
202	Unable to read item
223	Item already exists
235	Unable to update locked item
260	Unable to read locked item

For further information on using the Pick server, a sample Excel worksheet and Visual Basic application are included in the SAMPLES sub-directory.

Server Functions

AddItem method

`Server.AddItem file, ID, data`

Writes a new item to the Pick database. If the item already exists, error 223 is returned.

CallSub method

`Result = Server.CallSub(SubName [, function [, data]])`

Calls a Pick/BASIC subroutine (***SubName***) passing ***function*** and ***data*** as arguments. The Pick/BASIC subroutine must accept 4 arguments: Function, Data, ErrorCode, ErrorMessage. The returned value is stored in the Data argument before the subroutine returns. If an error occurs, the subroutine must set the ErrorCode argument to a non-zero value, and place the error message text in the ErrorMessage argument.

Connect method

`Result = Server.Connect([ServerName])`

Connects the server object to an AccuTerm session in server mode. If the optional ***ServerName*** is specified, then the object is connected to the specified server; otherwise, the first session which is in server mode is used. When establishing a connection, the ServerConnect function of the FTSERVER.SCR script is called. You can customize this script.

DeleteItem method

Server.DeleteItem *file*, *ID*

Deletes an item from the Pick database. If the item is locked by another process, error 235 is returned.

Disconnect method

Server.Disconnect

Disconnects the server object. When terminating the connection, the ServerDisconnect function of the FTSERVER.SCR script is called. You can customize this script.

Download method

Server.Download *SourceFile*, *SourceIDs*, *TargetFolder*
[, *Protocol* [, *Binary* [, *Overwrite*]]]

Uses the AccuTerm FT program to download files from the Pick host to the PC. *SourceFile* is the host file name; *SourceIDs* is a list of item-IDs, an asterisk (*) for all items, or an open parenthesis followed by the name of a saved list. Separate IDs with CR/LF; *TargetFolder* is the destination directory. *Protocol* is 1 for Kermit (default) or 0 for ASCII protocol. If *Binary* is zero (default) attribute marks are translated into CR/LF. *Overwrite* is non-zero to allow existing files to be overwritten (default is no overwrite).

Execute method

Result = Server.Execute(*command* [, *data* [, *capture*]])

Executes a TCL *command* on the host system. *Data* is passed as “stacked input” to the command. If *capture* is non-zero, the result of the command is returned, otherwise, the return value is empty.

Export method

Server.Export *SourceFile*, *SourceIDs*, *TargetFile*,
TargetFields [, *Header* [, *Explode* [, *Protocol* [, *Delimiter* [, *Overwrite*]]]]]

Uses the AccuTerm FTD program to download a file from the Pick host to the PC. *SourceFile* is the host file name. *SourceIDs* is a list of item-IDs, an asterisk (*) for all items, or an open parenthesis followed by the name of a saved list. Separate IDs with CR/LF; *TargetFile* is the destination file. If the destination file has a supported extension (.XLS, .WK1, .WKS, .WB1, .SYM, .DB2, .DBF), the resulting file will be exported in the native format for that file extension. *TargetFields* is a list of fields to include in the destination file; use asterisk (*) for all fields. Separate field names with

CR/LF. If **Header** is non-zero, the first line of the exported file will be a “header”. If **Explode** is non-zero, single values are repeated for items with multiple values. **Protocol** is 1 for Kermit (default) or 0 for ASCII protocol. If the destination file does not have a supported extension, then **delimiter** is used to specify Tab (0, default) or Comma (1) delimited ASCII file. **Overwrite** is non-zero to allow existing files to be overwritten (default is no overwrite).

IConv method

```
Result = Server.IConv(data, code)
```

Performs an “input conversion” on **data**. Conversion to be performed is **code**.

Import method

```
Server.Import SourceFile, TargetFile, reserved1,  
    Fields [, Header [, Skip [, Autold [, AutoldPrefix [,  
    AutoldStart [, Protocol [, Delimiter [, Overwrite  
    ]]]]]]
```

Uses the AccuTerm FTD program to upload a file from the PC to the Pick host. **SourceFile** is the PC file name; **TargetFile** is the destination file name. If the source file has a supported extension (.XLS, .WK1, .WKS, .WB1, .SYM, .DB2, .DBF), the file will be imported from the native format for that file extension. **Reserved1** is reserved for future use and should be passed as a null string. **Fields** is a list of fields to be imported (dictionary names or attribute numbers, or asterisk (*) for all fields). Separate field names with CR/LF. If **Header** is non-zero, the first line of the imported file is treated as a “header”. **Skip** is the number of “header” lines in the source file to skip. If **Autold** is non-zero, then the target item-IDs will be generated by concatenating **AutoldPrefix** to the item sequence starting with **AutoldStart**. **Protocol** is 1 for Kermit (default) or 0 for ASCII protocol. If the source file does not have a supported extension, then **delimiter** is used to specify Tab (0, default) or Comma (1) delimited ASCII file. **Overwrite** is non-zero to allow existing items to be overwritten (default is no overwrite).

OConv method

```
Result = Server.OConv(data, code)
```

Performs an “output conversion” on **data**. Conversion to be performed is **code**.

ReadItem method

```
Result = Server.ReadItem(file, ID [, attr [, value [,  
    subvalue [, locked ]]]] )
```

Reads an item from the Pick database and returns the item, attribute, value or sub-value. If the optional **locked** parameter is non-zero, the item is left locked after the read (same as Pick/BASIC READU statement). Error 260 is returned if the item was already locked by another process.

Readnext method

```
Result = Server.Readnext(file, attr [, value [,  
    subvalue ]])
```

Reads the next item from the Pick database using the current select list and returns the specified attribute, value or sub-value. If the list is exhausted, `LastError` will be set to -1.

UnlockItem method

```
Server.UnlockItem [ file, [, ID ]]
```

Unlocks an item locked by the ReadItem function. If **file** and **ID** are null, unlocks all items locked by the process.

Upload method

```
Server.Upload SourceFolder, SourceFiles, TargetFile  
[, Protocol [, Binary [, Overwrite ]]]
```

Uses the AccuTerm FT program to upload files from the PC to the host. **SourceFolder** is the PC directory where the files are uploaded from; **SourceFiles** is a list of file names separated by CR/LF. **TargetFile** is the destination file. **Protocol** is 1 for Kermit (default) or 0 for ASCII protocol. If **Binary** is zero (default) attribute marks are translated into CR/LF. **Overwrite** is non-zero to allow existing files to be overwritten (default is no overwrite).

WriteItem method

```
Server.WriteItem file, ID, data [, attr [, value [,  
    subvalue [, KeepLocked ]]]]
```

Writes an item, attribute, value or sub-value to the Pick database. If the optional **KeepLocked** parameter is non-zero, the item is left locked after the writing (same as Pick/BASIC WRITEU statement).

Standalone Kermit

A standalone version of the Kermit file transfer program is also included with AccuTerm. This is an implementation of industry standard Kermit in Pick BASIC. To use the standalone Kermit program, simply enter KERMIT at TCL.

To send from Pick to another Kermit, enter S to the first prompt, then enter the Pick file name and item list. Initiate the receive command on the remote Kermit.

To receive from another Kermit to Pick, enter R to the first prompt, then enter the Pick file name for the received items. Initiate the send command on the remote Kermit.

To change Kermit configuration settings, enter C to the first prompt. Most of these settings should not be changed from their default values, however, the "Source buffer index" may be tuned for best performance on your system, and "Use MCP conversion" may be disabled if the MCP conversion code does not function correctly on your system (Kermit sends will fail if the MCP conversion fails).

Integrating File Transfer Functions into Pick Applications

There are two ways to integrate AccuTerm' file transfer functions into your applications. First, and recommended, is to use the FT and FTD programs in PROCs or BASIC programs using the EXECUTE statement. Responses to all of the prompts may be "stacked" and fed to the file transfer utility.

Alternatively, two callable subroutines are provided for integrating AccuTerm' file transfer functions with your applications: DOSSVC and FTS.

DOSSVC performs three functions (each is optional). First, a "file" is sent to the DOS machine; next a DOS command is executed (this may be a BATCH file); finally, a file is retrieved from the DOS machine. The file to be sent, DOS file name, and DOS command are passed as arguments to the subroutine, and the received file is passed back to the calling program.

FTS performs a simple "file" transfer to or from the DOS machine. The "file" (a dynamic array) to be sent to (or received from) is passed as an argument to the subroutine. Complete interface information is included in the source code.

File Transfer Notes

R83 Systems

Older R83 based PICK systems require BASIC program files to be DC type. On those systems, you must edit the file D pointer and change the D to DC:

```
>ED MD FTBP
Top
.R/D/DC
001 DC
.FI
'FTBP' Filed.
```

G/A R91, Icon & CIE

It is necessary to enable control characters for BASIC programs to install or use the file transfer programs:

Type: CONTROL-CHARS ON to enable control characters.

McDonnell Douglas / Novadyne

To install the file transfer programs on a McDonnell Douglas system, use FTPINSMD.TXT for the file name in step 5 of the installation instructions. This is a custom installation script for McDonnell Douglas.

V/Mark UniVerse

It is necessary to inhibit case inversion to install and use the file transfer programs:

```
Type: PTERM CASE NOINVERT
```

To create the proper file type from an Ideal flavor account, use:

```
CREATE-FILE FTBP 1
```

from any other account flavor use:

```
CREATE-FILE FTBP 1 1,1,1
```

Certain control characters are used by the file transfer protocols, and must not be used for other system functions (retype line, kill, etc.) The reserved control characters are ASCII **SOH** (control-A), used by Kermit and ASCII **SUB** (control-Z), used as the end-of-file marker during ASCII file transfers.

Also, use FTPINSUV.TXT for the file name in step 5 of the installation instructions. This is a custom installation script for UniVerse.

UniData

To create the proper file type, use:

```
CREATE-FILE DIR FTBP
```

Certain control characters are used by the file transfer protocols, and must not be used for other system functions (retype line, kill, etc.) The reserved control characters are ASCII **SOH** (control-A), used by Kermit and ASCII **SUB** (control-Z), used as the end-of-file marker during ASCII file transfers.

Also, use FTPINSUD.TXT for the file name in step 5 of the installation instructions. This is a custom installation script for UniData.

jBase

To install the file transfer programs on jBase systems, use FTPINSJB.TXT for the file name in step 5 of the installation instructions. This is a custom installation script for jBase.

APPENDIX F

Object Bridge

Object Bridge is a Windows Object extension (OLE Automation) for Pick/BASIC programs. This allows a Pick/BASIC program to manipulate nearly any (OLE) Automation compatible object on the Windows machine. Examples of Automation objects include Word documents, Excel spreadsheets, and any public objects implemented using Visual Basic.

Object Bridge consists of three components: a Pick/BASIC interface implemented as a set of callable subroutines, a “remote procedure call” (RPC) conduit (AccuTerm 97 session) and a Windows object manager (installed with AccuTerm 97).

Object Bridge allows a Pick/BASIC program to create and destroy instances of Windows objects, set and retrieve property values from an object, invoke methods (subroutine calls and functions) on an object and process events produced by an object. The Pick/BASIC subroutines which provide this functionality are described below.

To use Object Bridge effectively, you may need some reference material in order to understand the object model of the application you are trying to control (Word, Excel, VB, etc). Also, the Object Browser tool available in Word or Excel (Macro -> Visual Basic Editor) or in the VB IDE is very useful.

The basic idea is this:

- 1) Initialize the Object Bridge environment
- 2) Create an object
- 3) Set property values, get property values or call subroutines or functions (invoke methods) on the object
- 4) Respond to desired events produced by the object
- 5) Release the object

Installing Object Bridge

Object Bridge requires the AccuTerm 97 Pick file transfer programs to operate. Install the file transfer programs per the instructions in Appendix E. Once the file transfer programs have been installed, create a program file for the Object Bridge subroutines (we recommend OBJBP, but any file name will work). Upload the Object Bridge subroutines using the FT utility. The files to upload are in the Atwin\PickBP\ObjBP folder. Compile and catalog the subroutines.

A sample program, CREATE.WORD.LETTER, for creating a Word letter document from a Pick/BASIC program is included in the SAMPLES folder.

Initializing the Object Bridge Environment

```
ATINITOBJMGR( ERRMSG, OPTS )
```

Call this subroutine before using any other Object Bridge routine. This verifies that a compatible version of AccuTerm 97 is connected, and initializes some important state information in the OPTS variable. You must maintain and pass the OPTS variable to all other Object Bridge routines.

If any of the Object Bridge routines encounter an error, the ERRMSG argument will be set to indicate the error, otherwise it will be null. Sometimes you can ignore an error, such as when attempting to set a property to an invalid value; other times you should call ATRESETOBJMGR to release all your objects and terminate the program.

Reset the Object Bridge Environment

```
ATRESETOBJMGR
```

This routine may be called if you want to release all objects and clean up the Windows object manager. This is useful if an error occurs and you want to abort all further use of any objects. You do not need to call ATRELEASEOBJECT if you call ATRESETOBJMGR.

Creating an Object

```
ATCREATEOBJECT( CLSID, OBJID, ERRMSG, OPTS )
```

This routine will attempt to create an object of the desired class (CLSID). Use the returned OBJID to refer to this object. When you no longer require access to the object, call ATRELEASEOBJECT passing the same OBJID, otherwise the object remains present in Windows memory.

Releasing an Object

```
ATRELEASEOBJECT( OBJID, ERRMSG, OPTS )
```

This routine releases the specified object, as well as any objects "derived" from the specified object (OBJID). A dereived object may be an object returned by a call to ATGETPROPERTY or ATINVOKEMETHOD when the return value is itself another object.

Setting Object Properties

```
ATSETPROPERTY( OBJID, PROPNames, PROPVALUES,
               ERRMSG, OPTS )
```

Use this routine to get the values of one or more properties of an object. Specify the property names you desire in the PROPNames argument, separated by attribute marks. The corresponding property values to set are passed in the PROPVALUES argument, also separated by attribute marks. If an error occurs, the error message will be in the corresponding attribute of the ERRMSG argument.

Getting Object Properties

```
ATGETPROPERTY( OBJID, PROPNames, PROPVALUES,
               ERRMSG, OPTS )
```

Use this routine to set the values of one or more properties of an object. Specify the property names you desire in the PROPNames argument, separated by attribute marks. The corresponding property values are returned in the PROPVALUES argument, also separated by attribute marks.

If an error occurs, the error message will be in the corresponding attribute of the ERRMSG argument.

Some properties are actually objects themselves. This is a "derived" object. In this case, you can use the returned property value as an object ID in other Object Bridge calls. When the object which returned the "derived" object is released, all of its derived objects will also be released. Reassign the derived object will not release its creator.

Invoking Methods (calling subroutines and functions)

```
ATINVOKEMETHOD( OBJID, METHNAMES,
METHARGS, RESULT, ERRMSG, OPTS )
```

Call this routine to invoke a "method" on the object. A method is a subroutine or function. A subroutine will not return a value (RESULT will be null). A function will return a result in the RESULT argument. Multiple methods may be called. Separate method names in the METHNAMES argument by attribute marks. If a method requires arguments, the arguments are passed in the METHARGS parameter, with arguments for each method separated by attribute marks. Within the method arguments for one method, multiple arguments are separated by value marks. Results from multiple methods are returned in the RESULTS argument separated by attribute marks.

Some methods return objects. In this case, you can use the returned value as an object ID in other Object Bridge calls. When the object which returned the "derived" object is released, all of its derived objects will also be released. Releasing the derived object will not release its creator.

Enabling Event Processing

```
ATSETEVENT( OBJID, EVENTNAMES, ENABLE, ERRMSG,
OPTS )
```

Many objects support events, which notify the object's owner of certain actions, such as closing the program or clicking a button. Object Bridge only notifies the Pick application of those events which have been enabled by calling ATSETEVENT. Specify the event names in the EVENTNAMES arguments, separated by attribute marks. Specify whether the event is

enabled (1) or disabled (0) in the corresponding `ENABLE` argument, also separated by attribute marks.

Polling for Events

```
ATGETEVENT(OBJID, EVENTNAME, ARGS, TIMEOUT,
ERRMSG, OPTS)
```

Call this routine to wait for an event from any objects for which you have enabled events. You do not pass `OBJID` to this routine, it is passed back to your program when an event occurs so you can identify the source of the event. `EVENTNAME` and `ARGS` are also passed back to your routine. Some events allow you to modify one or more of their arguments, for example, to cancel an operation. You must preserve `OBJID`, `EVENTNAME` and `ARGS` (except for modified values in `ARGS`) when processing the event so they can be passed to `ATENDEVENT` when you complete processing the event. Event argument values are separated by value marks. You must always call `ATENDEVENT` after processing the event. You can call other routines in between, but no more events can be processed, and the Windows object may be roadblocked until you call `ATENDEVENT`.

If you want `ATGETEVENT` to return to your program even if an event did not occur, you can specify the number of milliseconds to wait for the event in the `TIMEOUT` parameter. Set `TIMEOUT` to zero to wait forever for the event.

Completing Event Processing

```
ATENDEVENT(OBJID, EVENTNAME, ARGS, ERRMSG,
OPTS)
```

Call this routine after handling an event. Pass back the `OBJID`, `EVENTNAME` and `ARGS` as they were returned by the `ATGETEVENT` routine, except for any argument values you have modified.

You must call this routine after processing an event returned by `ATGETEVENT`, otherwise no other events may be processed and the Windows application which implements the object may not be able to terminate.

APPENDIX G

Custom Features

AccuTerm 97 supports several custom options which may be specified in the ATWIN.INI file. These options are never updated by the **Tools Settings** dialog; they must be entered directly in the ATWIN.INI file using NOTEPAD (or another text editor).

Version Verification

The automatic version verification feature of AccuTerm 97 may be disabled by adding the line `NoVersionVerify=True` to the `[Options]` section of ATWIN.INI. This is not recommended, but may be necessary in those cases where the versions of the OCX and DLL files are known to be mismatched, but do not cause any problems.

Specific OCX and DLL files can be excluded from the verification process by modifying the `[Components32]` section of ATWIN.INI. Change the value of the excluded file from the normal size/time/date value to **Ignore**. For example, to exclude `MSVBVM50.DLL` from the verification process, modify the entry for `MSVBVM50.DLL` to read:
`MSVBVM50.DLL=Ignore.`

Customizing the Installation Process

The AccuTerm 97 installation program will install vendor supplied files when placed in the VARFILES directory of the last diskette. Files found in this directory will be copied into the target ATWIN directory. Files found in VARFILES\WINDOWS will be copied to the user's WINDOWS directory, and files found in VARFILES\SYSTEM will be copied to the user's WINDOWS\SYSTEM directory. Files found in VARFILES\HOME will be copied to the "home" AccuTerm/Windows directory (usually this is the same as the target ATWIN directory, but in a network installation, the

ATWIN directory may be shared, but the "home" directory will be private). This is where a customized ATWIN.INI, .LYT or .CFG file should be located.

The AccuTerm 97 installation program can also run a vendor supplied installation program. The vendor supplied program must be named "varsetup.exe" and must reside in the same directory as the main AccuTerm setup program.

The AccuTerm 97 installation can be run in an automatic mode. Default values for several installation parameters may be specified in the SETUP.INI file which must reside in the same directory as the main AccuTerm setup program. The SETUP.INI file can be used to specify default values for several installation prompts:

```
[Defaults]
Maindir= program installation directory
Homedir= users home directory
Group= start menu group
Options=A
```

The **Maindir** is the directory where the program files are installed. If this is on a network share, specify the **Homedir** as a directory where the user's configuration files will be stored. The **Group** is the start menu group for the shortcut to start AccuTerm. The **Options=A** is included if you want the enhanced image support loaded (omit this line if standard image support is desired).

If the setup program is run with the /S or /Q command line options, the installation is performed using the default values from the SETUP.INI file. The /Q option displays the installation progress; the /S option is completely silent.

Custom Icon and Title

It is possible to use a custom icon and / or window title with AccuTerm 97. To change the icon, add an **Icon** item to the [Custom] section of ATWIN.INI. To change the window title, add a **Title** item:

```
...
[Custom]
Icon= filename
Title= new window title
...
```


Custom Hot Key Blocking

It is possible to block the action of any hot keys, except those hot keys which open a top-level menu (use the menu designer to customize the menu if you need to modify this action). To enable this feature, add a **BlockedHotKeys** item to the **[Custom]** section of ATWIN.INI. Assign the list of hot keys to block to this item. For example, to block the ALT+D (DOS prompt) and ALT+P (printer toggle) hot keys, add **BlockedHotKeys=DP** to the **[Custom]** section of ATWIN.INI.

Custom Mouse Action

AccuTerm supports the use of a Mouse Pattern Table. The Mouse Pattern Table associates patterns of text on the screen with response strings. The response strings can either be sent to the host or executed as macro commands. Each entry specified a pattern, mouse button, click response and double click response.

AccuTerm's host-controlled mouse functions override the Mouse Pattern Table, and if the host has enabled the mouse using the **ESC STX 1** command, then the Mouse Pattern Table functions are disabled until the host disables the mouse using the **ESC STX 0** command.

The Mouse Pattern Table is specified in the **[Custom]** section of the ATWIN.INI file. The entry is:

```
...  
[Custom]  
MouseTable=filename  
...
```

The Mouse Pattern Table file (**filename**) is a tab delimited text file (ANSI character set.) Each line in the file is an entry in the Mouse Pattern Table. The default directory for this file is the ATWIN home directory.

Note: the DOS editor does not preserve tab characters when saving a file, so edit this file using Notepad instead!

Besides the ability to read the Mouse Pattern Table from a file, the table can

also be downloaded from the host, and therefore, changed on the fly as applications have different mouse requirements. To download the table, use the following AccuTerm private escape sequence:

ESC STX h CR	clears the Mouse Pattern Table
ESC STX h <i>entry</i> CR	adds <i>entry</i> to the Mouse Pattern Table

Each entry is made up of four fields separated by tab (HT) characters:

button HT pattern HT click HT dblclk

The Mouse Pattern Table can contain up to 128 entries. Each entry specifies the mouse button desired (***button***), the pattern (***pattern***), the response to a click (***click***) and the response to a double click (***dblclk***). The fields of each entry are separated by a tab character. The mouse button is a single digit: 1 = left, 2 = right, 3 = middle. Pattern and response strings are described below.

Patterns

The pattern field is a "regular expression", similar to the regular expressions used by the Unix **grep** command:

.	matches any character
^	beginning of line
\$	end of line
[<i>list</i>]	any characters in <i>list</i> (<i>list</i> can include a range of characters such as 0-9)
[^ <i>list</i>]	any characters not in <i>list</i>
:A	any alpha character
:D	digit (0 - 9)
:N	any alpha or numeric character
?	zero or one of the preceding item
*	zero or more of the preceding item
+	one or more of the preceding item
{	marker for the beginning of the word to return if the pattern matches
}	marker for the end of the word to return if the pattern matches
\	causes the next character in the pattern string to be used literally; for example, \. matches a dot
<i>any other char</i>	matches itself

Responses

Reponses are encoded using the function key format (see **Keyboard Settings**). Just like function keys, if the response is a string enclosed in square brackets ([]), the response is a "macro" command, and it is executed, rather than being sent to the host (see **Scripting**). Note: if the macro begins with an asterisk (*), it causes a "context menu" to be displayed, rather than executing as a macro. This can be used with a customized popup menu, described in the next section. For example, a response of [*reports] will open the popup menu named "reports".

Certain special tokens are permitted in the response string which are useful for returning information to the host, or for passing as arguments to a macro command. These special tokens are:

%WORD%	replaced by the "word" which matched the pattern.
%COL%	replaced by the zero-based text column where the mouse was clicked.
%ROW%	replaced by the zero-based text row where the

	mouse was clicked.
%BEG%	replaced by the zero-based text column where the matched text begins.
%END%	replaced by the zero-based text column where the matched text ends.
%MOVE%	replaced by the cursor movement commands required to position the cursor to the location of the mouse click.
%%	replaced by the percent (%) character.

Customizing the Menu and Toolbar

AccuTerm 97 includes a menu design utility which can be used to customize AccuTerm's main menu, popup menu, toolbar and status line. *Use caution when modifying AccuTerm's menu structure; it is possible to make AccuTerm non-functional.* The menu designer allows you to add, remove and modify menu items, sub-menus, toolbar buttons and status line fields.

Menu Structure

The menu file stores the complete menu structure used by AccuTerm. The menu structure is hierarchical, and consists of zero or more top-level menu items: Menu Bar, ToolBar, Status Line, and Popup Menu. You can remove any of these items, or add new ones as desired (it is possible to have more than one main Menu Bar, ToolBar, Status Line or Popup Menu). Each top-level menu item contains "tools". Each "tool" can specify an action, or open a sub-menu. There are built-in menu tools, one for each of the menu or toolbar actions included on AccuTerm's default menus. You can modify the built-in tools, but they cannot be deleted from the tool library. You can create your own tools as well.

Main Menu

The main menu is displayed when AccuTerm 97 is first started, or when all session windows have been closed. This menu is unique in that no session-specific items are included. As soon as a session is opened, the main menu is replaced by the session menu. AccuTerm 97 loads the main menu from the **dfltmmain.mnu** file. If you want to modify the main menu, you should modify this file. If you need to restore the main menu to its original state, simply copy the **stdmain.mnu** file over **dfltmmain.mnu**.

Session Menu

Once a session is open, AccuTerm 97 replaces the main menu with the session menu. By default, AccuTerm 97 loads the session menu from the **dfltsess.mnu** file. This is the menu which is displayed when a new session is created. Optionally, a **.CFG** file can specify an alternate menu to use when the session is opened. Use **File Update Session** from the menu designer to modify the **.CFG** file with an alternate session menu. If you need to restore the default session menu to its original state, simply copy the **stdsess.mnu** file over **dfltsess.mnu**.

Menu Bar

The Menu Bar is the standard Windows menu, displayed at the top of AccuTerm's main window. Normally, items on this menu open up sub-menus; for example, clicking on the Menu Bar **File** menu item opens up the File sub-menu, which contains tools for New, Open, Save, etc. Properties of the Menu Bar are Name, Caption, Visible and DockingArea.

Name: name used to reference this menu item.

Caption: displayed only if the Menu Bar is "floating".

Visible: check this box if the Menu Bar is visible.

DockingArea: 1 if Menu Bar is "docked" to the top of the window (default), 2 if bottom, 4 if left side, 8 if right side.

ToolBar or StatusLine

The ToolBar is the button bar displayed just below the Menu Bar at the top of AccuTerm's main window. Normally, items on the ToolBar are "tools" with icons, but without captions, which perform some action when clicked. The default tool bar has buttons for New, Open, Save, etc. You can create your own tools, and add them to the ToolBar.

The Status Line displays informaton about the state of AccuTerm at the bottom of AccuTerm's main window. Normally, items on the Status Line are "tools" with icons and caption, but which do not perform any action when clicked. The default status line has tools for the current cursor position, screen size, emulation, etc.

Properties of the ToolBar / Status Line are Name, Caption, Visible, and DockingArea.

Name: name used to reference this menu item.

Caption: displayed only if the ToolBar or Status Line is “floating”.

Visible: check this box if the item is visible.

DockingArea: 1 if the item is “docked” to the top of the window (default for ToolBar items), 2 if “docked” to the bottom of the window (default for Status Line items), 4 if left side, 8 if right side.

Popup Menu

The Popup Menu (context menu) is displayed when the user clicks the AccuTerm screen with the right mouse button. Normally, items on this menu perform some action, like Copy, Paste, etc. You can use the **Custom Mouse Action** (described earlier in this section) to open your own popup menu when the mouse is clicked over a specified character pattern. Properties of the Popup Menu are Name, Caption and Visible.

Name: name used to reference this Popup Menu.

Caption: not used in Popup Menus.

Visible: check this box if the item is visible.

Sub-Menu

A sub-menu can be attached to any of the top-level menu items (Menu Bar, ToolBar, Status Line, Popup Menu). A sub-menu is opened when the user clicks its caption or button on its parent menu. The sub-menu can contain other sub-menus, or “tools” which perform some action. Properties of the Sub-Menu are Name, Caption, Visible and BeginGroup.

Name: name used to reference this menu item.

Caption: text is displayed on the menu or toolbar. For menu items, one letter of the caption may be marked as a “hot key” by preceding that letter with an ampersand (&).

Visible: check this box if the sub-menu is visible.

Begin Group: check this box to place a separator between this item and the tool preceding it in the parent menu.

Tool

A “tool” is a menu item which performs some action. A tool has a Caption, Tool Tip Text, Action, and optional Icon. A tool which performs an action acts like a button (the Button check-box is checked). Tools on the Status Line usually do not perform any action, so they do not have the Button check-box checked. Other tool properties are Height, Width, Begin Group, Visible, and Enabled.

Caption: text is displayed on the menu or toolbar. For menu items, one letter of the caption may be marked as a “hot key” by preceding that letter with an ampersand (&).

Tool Tip Text: text displayed in a floating window when the mouse is paused over the menu item.

Action: macro name to execute enclosed in square brackets ([]) or character sequence to transmit to the host. Control characters are entered as described in “function key programming”. If this field is blank, the default action for the built-in tool is performed.

Height: fixed height in pixels. Leave this field blank for auto-height.

Width: fixed width in pixels. Leave this field blank for auto-width.

Icon: the icon displayed in the menu, toolbar or status line. Click Select Icon to select a new icon; click Clear Icon to remove the icon from the tool.

Begin Group: check this box to place a separator between this tool and the tool preceding it in the parent menu.

Visible: check this box if the tool is to appear in the menu, toolbar or status line. For built-in tools, this property may change to reflect AccuTerm's current state.

Enabled: check this box if the tool is to be "enabled". If the tool is not enabled, it is displayed in a light-grey color, and no action is performed when the tool is clicked. For built-in tools, this property may change to reflect AccuTerm's current state.

Checked: check this box to indicate that the tool is "checked". When the tool is checked, a check-mark will be displayed next to the tool caption instead of an icon. For built-in tools, this property may change to reflect AccuTerm's current state.

Using the Menu Designer

To start the menu designer, locate the **atmnudsg.exe** file (in \Program Files\Atwin) using Windows Explorer, and double click on the file name. Use **File Open** to open an existing menu file, then **File Save As** to save the menu under a new name.

After a menu file is opened in the menu designer, two panels are displayed: the Menu Preview, and the Menu Designer. The Menu Preview panel displays a sample screen showing the current menu file. You can exercise the menu system using this panel. The Menu Designer panel is used to manipulate the menu structure.

The menu structure is displayed as a hierarchy, with AccuTerm as the root. The top-level menu items Menu Bar, ToolBar, Status Line and Popup Menu are displayed at the next level. Under the top-level menu items are sub-menus or tools. Sub-menus may contain other sub-menus or tools.

To expand the menu display, click on the plus sign next to the item you want expanded. To collapse the display, click on the minus sign next to the item. To insert a new item into the structure, click the location where you want the new item inserted with the right mouse button, and select Add. To remove an item, click the item with the right mouse button, and select Remove. To modify an item, click the item with the right mouse button and select Properties.

When you are finished modifying the menu structure, use **File Save** to save the menu. To update a session configuration file, use **File Update Session**.

LANPAR Emulation

AccuTerm's VT220 emulation can be modified to emulate the LANPAR Vision II terminal by adding the `Lanpar=True` item to the [Custom] section of ATWIN.INI. When LANPAR emulation is enabled, the differences to VT220 are:

Function Keys: `CSI 2 ; key ; bank ; 0u` may be used to program function keys. `CSI 0 ; key ; bank u` may be used to execute a function key, and `CSI 4 ; key ; bank.u` may be used to clear function key programming. **Key** is the function key number (1 - 12), and **bank** selects the shift status (1 = normal, 2 = shifted, 3 = control, 4 = control+shift, 5 = alt, 6 = alt+shift). The programming command must be followed by a string which contains the program data in the form of: **delimiter switch route data delimiter**, where **delimiter** is a single ASCII character delimiter, **switch** is a "routing switch character", **route** is 0 (ignore data), 1 or 2 (program data to function key), **data** is a string of characters, including control characters to be programmed into the key, and **delimiter** is the terminating delimiter.

Message Window: AccuTerm only supports a single line host message, which is always displayed on the last line of the screen. `CSI 2 v` may be used to position the cursor in the message line, and clear the line.

`CSI 1 v` may be used to position the cursor in the message line. `CSI 0 v` causes the cursor to leave the message line and return to the current screen. `CSI 3 ; n v` hides ($n = 0$) or displays ($n > 0$) the message line.

Screen Pages: AccuTerm supports up to 25 pages. Under LANPAR emulation, `CSI 1 ; 0 ; page p` will change to page **page**. `CSI 1 ; 1 p` will change to the next page. `CSI 1 ; 2 p` will change to the previous page.

Miscellaneous: `ESC # 7` prints the screen.

Index

- AccuTerm Object Reference, 79
- ADDS Programming, 153
 - cursor positioning, 154
 - erasing and editing, 155
 - line graphics, 157
 - operating modes, 153
 - printer control, 158
 - video attributes, 156
- ADDS Viewpoint 60, 153
- ADDS Viewpoint A2, 153
- ADDS Viewpoint Enhanced, 137
- ADDS Viewpoint Tables, 185
- ANSI, 40
- ANSI Attribute Code Table, 189
- ANSI Function Key Table, 190
- ANSI Key Code Table, 191
- ANSI Programming, 159
 - character set selection, 163
 - cursor positioning, 165
 - erasing and editing, 167
 - function key programming, 170
 - operating modes, 161
 - printer control, 169
 - video attributes, 169
- ANSI Tables, 189
- Answerback, 38
- application object, 79
- ASCII, 60
- ASCII Codes, 193
- ASCII settings, 23
- Auto Answer, 30
- Auto Close, 21, 26
- Auto Dialer, 55
- Auto-convert, 203
- auxiliary port, 67
- Baud Rate, 28
- Bell sound, 53
- binary mode, 62, 65
- bitmap files, 135
- Blinking, 42
- Capture Source, 68
- checksum, 24
- color, 45
- Comm Errors, 26
- Comm Port, 27
- command line, 11
- Command Line, 10
- commands
 - private, 131
- Connection Wizard, 10
- CRC, 24
- Cursor shape, 42
- Custom Features, 231
- Custom Icon and Title, 232, 233
- Custom Mouse Action, 233
- Customizing the Menu and Toolbar, 236
- Data Bits, 28
- data capture
 - starting, 67, 133
 - stopping, 133
- Data Capture, 67
- data compression, 23
- Database files, 203
- Default capture directory, 21
- Default download directory, 24
- default phone book, 20
- default transfer mode, 24
- Delimited files, 203
- Device, 26
- dial, 56
- Dialing Options, 31
- Directories, 8
- DOSSVC, 197
- Download, 59, 64
 - host control of, 132
- Dtr/Dsr, 28
- Duplex, 26
- Edit Menu, 13
- eighth bit
 - quoting, 23
 - stripping, 26
- emulations, 37
- environment variable, 8
- error checking

- checksum, 24
- CRC, 24
- error detection, 60
- execute DOS command, 131
- existing files, 63, 65
- Exporting files, 203
- extended mode, 40, 133
- Extended Video, 42
- extra row, 40
- File Menu, 12
- File Selection, 60, 65
- file transfer
 - host control of, 132
 - Pick utilities, 197
 - status of, 132
- File Transfer, 59
 - Pick Server, 214
- file transfer protocol
 - ASCII, 59, 132
 - Kermit, 59, 132, 197
 - overwrite, 132
 - resuming Zmodem file
 - transfer, 132
 - Xmodem, 59, 132
 - Ymodem, 59, 132
 - Zmodem, 59, 132
- file transfer protocol options, 22
- File transfer, Direct Pick-to-Pick
 - transfer, 209
- File transfer, DOS file to Pick
 - file, 206
- File transfer, DOSSVC
 - Subroutine, 221
- File transfer, Exporting Pick
 - data, 205
- File transfer, FT options, 202
- File transfer, FT utility, 200
- File transfer, FTD options, 208
- File transfer, FTD utility, 203
- File transfer, FTPICK options,
 - 213
- File transfer, FTPICK utility, 209
- File transfer, FTS Subroutine,
 - 221
- File transfer, Importing data, 206
- File transfer, Indirect Pick-to-Pick
 - transfer, 210
- File transfer, Integration with
 - applications, 221
- File transfer, Kermit program,
 - 221
- File transfer, Pick file to DOS
 - file, 205
- File transfer, Pick items to DOS
 - files, 200
- Font, 43
 - legible, 44
 - scaling, 44
 - size, 43
 - style, 44
- FT, 197
- FTD, 197
- FTPICK, 197
- FTS, 197
- FTSERVER, 214
- FTSETUP, 214
- function key labels, 40
- function key programming, 48
- function keys
 - programming, 134
- garbled data, 31
- Handshake, 28
- Hangup, 56
- Hardware Requirements, 3
- Help, 16
- Help Menu, 16
- History buffer, 40
- History Rows, 40
- Host Directed File Transfer, 66
- Icon, 42
- image display
 - host control, 135
- Importing files, 203
- Installation, 5
- Integrating Pick and DOS, 197

- inter-line delay, 60
- interrupts, 28
- jBase, 223
- JPEG files, 135
- Kermit, 60, 197
- Kermit settings, 23
- Key Types, 48
- Keyboard Commands, 17
- LANPAR Emulation, 241
- license, 16
 - determining license type, 133
- line end
 - ASCII, 23
- macro commands
 - host control, 134
- McDonnell Douglas, 222
- MDI, 1, 9
- metafiles, 135
- MIDI sound files, 135
- Minimize, 42
- Modem, 30
- mouse
 - host control of, 131
- Mouse
 - enabling, 42
- Mouse Pattern Table, 233
- mouse wheel, 40
- Multiple Document Interface, 1, 9
- network
 - Modular Software PicLan, 231
 - PicLan, 231
 - Host, 35
 - TCP/IP, 231
 - Telnet, 231
 - Host, 33
- Network
 - Telnet
 - Alternate break option, 34
 - Binary option, 34
 - Bypass option, 34
 - Echo option, 33
 - Options, 33
 - Port, 33
 - Terminal Type, 34
 - Timeout, 33
- Network Installation, 7
- normal mode, 40, 133
- Novadyne, 222
- Object Bridge, 225
- object hierarchy, 79
- Object Reference, 79
 - AccuTerm object, 81
 - Creating new session, 88
 - Events, 80
 - Menu object, 126
 - Methods, 80
 - MnuBand object, 127
 - MnuTool object, 128
 - Predefined session objects, 87
 - Properties, 80
 - ScreenBlock object, 125
 - Server object, 125
 - Session object, 87, 88
 - Sessions collection, 86
 - Settings object, 114
 - Type Library, 80
- Optional File Installation, 231
- Parity, 28
- Paste, 23
- Paste Options
 - line end, 52
 - paste end, 52
- phone book file, 20
- Phone directory, 55
- phone directory file, 57
- Pick File Transfer Programs
 - installation, 198
- Pick PC Console Programming, 173
 - cursor positioning, 173
 - erasing and editing, 174

- operating modes, 173
- printer control, 177
- protect mode, 176
- video attributes, 175
- Pick PC Monitor, 37, 40
- Pick Server, 214
- Pick/BASIC, 66
- PicLan, 35
- Printer
 - job close timeout, 51
- Printer Selection, 50
- Private Commands, 131
- Procomm VP60, 153
- programming
 - AccuTerm, 131
- Protocol, 60, 65
- release, 133
- Reset Menu, 16
- Rts/Cts, 29
- Screen Auto Forward, 42
- Screen Print, 50
- Screen Sizes, 40
- Script
 - AppActivate statement, 74
 - AppClose statement, 74
 - AppFind function, 74
 - AppGetActive function, 74
 - AppGetPosition statement, 75
 - AppGetState function, 75
 - AppHide statement, 75
 - AppList statement, 75
 - AppMaximize statement, 75
 - AppMinimize statement, 75
 - AppMove statement, 75
 - AppRestore statement, 75
 - AppSetState statement, 76
 - AppShow statement, 76
 - AppSize statement, 76
 - Chain statement, 76
 - closing, 71
 - controlling AccuTerm, 73
 - creating, 69
 - debugging, 73
 - Debugging, 77
 - editing, 71
 - FileExists function, 77
 - Item function, 77
 - ItemCount function, 77
 - Language Extensions, 74
 - Line function, 77
 - LineCount function, 77
 - loading, 70
 - OpenFileName function, 77
 - Pause statement, 78
 - printing, 71
 - Random function, 78
 - running, 71
 - running from command line, 71
 - running from function key, 72
 - running from host computer, 72
 - running from menu or toolbar, 72
 - saved in layout file, 72
 - SaveFileName function, 78
 - saving, 70
 - Word function, 78
 - WordCount function, 78
- Scripting, 69, 79
- Scripting language, 2
- scroll bars, 40
- Scrollbar, 40
- selection, 13
- serial number, 16, 133
- Serial Port Device, 27
- Server Functions, 217
- Session Title, 39
- SET command, 8
- Settings
 - Colors, 45
 - File Transfer, 22
 - General, 20
 - Keyboard, 47

- Miscellaneous, 52
- Printer, 49
- Serial Port Device, 27
- Session, 25
- Terminal Type, 37
- Setup, 5
- shortcut, 11
- Single instance, 21
- Slave Printer, 50
- Software Requirements, 3
- sounds
 - host control, 135
- Spreadsheet files, 203
- start capturing, 67
- Stop Bits, 28
- Strip Eighth Bit, 26
- System Requirements, 3
- TAPI, 30
- TARGA files, 135
- TCP/IP, 32
- TCP/IP hosts file, 21
- Tektronix, 37
- Telnet, 32
- Terminal Emulation, 37
- text mode, 62, 65
- TIFF files, 135
- Timeout, X/Y/Zmodem, 23
- Tools Menu, 15
- Underline, 42
- UniData, 223
- UniVerse, 223
- Upload, 59
 - host control of, 132
- Using the Menu Designer, 240
- VBA, 2, 69
- Version Verification, 231
- Video Pages, 40
- View Menu, 14
- Viewpoint, 37
- Viewpoint 60, 153
- Viewpoint A2, 153
- Viewpoint Attribute Code Table, 186
- Viewpoint Cursor Address Table, 185
- Viewpoint Key Code Table, 187
- Viewpoint Tables, 185
- Visual Basic for Applications, 69
- VT-220, 40
- VT-220 Programming, 159
- VT-52 Programming, 171
- Wave sound files, 135
- wheel mouse, 40
- WIN.INI file, 8
- Window Menu, 16
- Wyse, 40
- Wyse 50, 137
- Wyse 60, 137
- Wyse Attribute Code Table, 180
- Wyse Cursor Address Table, 179
- Wyse Function Key Table, 182
- Wyse Graphic Character Table, 181
- Wyse Key Code Table, 183
- Wyse Programming, 137
 - attributes, 146
 - cursor positioning, 141
 - erasing and editing, 144
 - function key programming, 151
 - line graphics, 148
 - operating modes, 137
 - printer control, 149
 - protect mode, 147
- Wyse Tables, 179
- Xmodem, 60
- Xon/Xoff, 28
- Ymodem, 60

Zmodem, 60
 crash recovery, 63
Zmodem, automatic download,
 24

